

Gerçek Zamanlı Nesne Tanıma Uygulamaları için HOG Algoritmasının FPGA Tabanlı Gömülü Sistem Uyarlaması

Implementation of HOG algorithm for Real Time Object Recognition Applications on FPGA based Embedded System

Fuat Karakaya, Halis Altun ve Mehmet Ali Çavuşlu

Niğde Üniversitesi Elektrik Elektronik Mühendisliği

alicavuslu@yahoo.com, {halisaltun,fkarakaya}@nigde.edu.tr

Özetçe

HOG algoritması son yıllarda kompleks içerikli görüntülerde nesne tanıma işlevini yüksek başarımla gerçekleştirebilen bir algoritma olarak kullanılmaktadır. Bu basit fakat etkin tanımlayıcının donanımsal gerçekleştirilmesi bir çok temel örüntü tanıma işlevinde kullanılabilirliği söz konusu olduğundan önemlidir. Bu çalışmada HOG algoritmasının, farklı geometrik şekilleri yüksek bir başarımla tanıyabilen donanımsal uyarlaması FPGA üzerinde gerçekleştirilmiştir. İncelenen görüntü içerisinde geometrik şeklin yatay ve dikey kenar bulma algoritmaları kullanılarak gradyan büyüklük ve açıları hesaplanmıştır. Elde edilen sonuçlar histogram üretmek için kullanılmış ve bu histogramların her bir şekil için ayırt edici karakteristiğe sahip olduğu gözlemlenmiştir. Kural tabanlı bir sınıflandırıcı uyarlanarak gerçek zamanlı olarak nesne tanıma uygulamalarında kullanılabilecek bir gömülü sistem yaklaşımı başarı ile sunulmuştur.

Abstract

Recent years HOG algorithm has been used to recognize objects in images, with complex content, with a very high success rate. Hardware implementation of this algorithm is very important because of the fact that it can be used in many object recognition applications. In this work HOG algorithm is implemented on FPGA to recognize different geometrical figures with a very high success rate. Objects vertical and horizontal edges have been sharpened using edge detection algorithms to calculate magnitude and angle of the local gradients. Obtained result are used to construct the histograms of gradient orientation. It is observed that each constructed histogram have distinctive features for every object. Rule based classifiers has been used to implement a successful real time object recognition approach on embedded system.

1. Giriş

Nesne sınıflandırma ve tanıma, çok farklı endüstriyel ve askeri uygulamaları olması açısından literatürde yoğun olarak çalışılmış bir problemidir. Ancak bu alan, son birkaç on yıl içerisinde gerçekleşen tüm gelişmelere rağmen halen zor ve çözülmesi gereken problemler içermektedir. Literatürde son yıllarda nesne sınıflandırma ve tanıma problemin çözümünde basit fakat etkin tanımlayıcıların önerildiği görülmektedir. Histogram of Oriented Gradient- (HOG) gibi bu tanımlayıcılar farklı koşullar altında yüksek başarımla vermektedir. HOG, ilk

olarak Shashua ve diğerleri [1] tarafından yaya tanıma sistemlerinde kullanılabilecek tanımlayıcılar olarak önerilmiştir. Dalal ve diğerleri [2] bu yeni tanımlayıcıları başarı ile kompleks ortamlarda insan tanıma problemine uyarladılar. Bu basit fakat etkin tanımlayıcı, başarılı uygulamalarından dolayı, literatürde yoğun bir ilgi kazanmış ve bir çok uygulamada kullanılmaya başlanmıştır [3].

Bu gelişmelere paralel olarak yarı iletken teknolojilerinde de son yıllarda yaşanan gelişmeler işaret işlem unitelerini, hafıza birimlerini ve farklı sensörleri içeren yapıların tek bir gömülü sistem üzerinde uyarlanmasını mümkün kılmaktadır. Bu tür bir gelişme önceki yıllarda pratik olarak gerçekleştirilmesi mümkün olmayan yeni uygulamaların donanım üzerinde gerçekleştirilebilmesine imkân sağlamıştır. Bu uygulamalar gerçek zamanlı plaka okuma [4], yüz tanıma [5], trafik kazalarını önlemek için sürücü yorgunluğunu belirleme [6] temel görüntü işlem uygulamaları [7-9], görüntü işlem uygulamalarında kullanılacak sınıflandırıcıların uyarlanması [10-14] şeklinde sıralanabilir

Genel olarak obje sınıflandırmada özel olarak ise yüz tanıma problemlerinde gerçek zamanlı çözümler sunabilmek için FPGA tabanlı bir YSA sınıflandırıcı uygulaması Smach ve diğerleri tarafından önerilmiştir [5]. Önerilen yaklaşımda aktivasyon fonksiyonunun donanım üzerinde gerçekleştirilmesinde yaşanan zorluklar fonksiyonun lineerleştirilmesi ile basitleştirilmiştir. Yapay sinir ağında ağırlıklı bağlantılar 8 bitlik sabit nokta ağırlıklar olarak ele alınmış ve FPGA olarak Xilinx firmasının 52 MHz saat hızına sahip VERTEX V1000BG960 ürünü kullanılmıştır. Polinomial çekirdek fonksiyon kullanımı FPGA uygulamasında sağlayacağı avantajdan dolayı öngörülmüştür. Xilinx Virtex XCV1000 FPGA modülü kullanarak destek vektörleri ile giriş vektörleri arasındaki iç çarpım (dot-product) bu polinom tarafından belirlenmektedir. FPGA tabanlı DVM kullanılarak bir nesne bulma ve sınıflandırma problemi Reyna ve diğerleri [15] tarafından önerilmiştir. Aguita ve arkadaşları [12] ise En Küçük Kareler (Least Squared) DVM algoritmasının FPGA tabanlı uyarlamasını Xilinx Virtex II V2MB1000 geliştirme boardu kullanarak gerçekleştirmişlerdir

Yukarıda literatürde sıralanan gerçek zamanlı FPGA tabanlı nesne sınıflandırma ve tanıma uygulamalarının bir uzantısı olarak, bu çalışmada, basit ancak güçlü ve paralelleştirmeye müsait bir tanımlayıcı olması açısından HOG tanımlayıcısının FPGA üzerinde donanımsal olarak uyarlaması önerilmektedir. Bu çalışma ile oldukça başarılı

uygulamaları söz konusu olabilecek olan HOG tanımlayıcıların alan programlanabilir kapı diziler üzerinde nesne tanıma uygulamalarında kullanılabilecek bir donanımsal uyarlamasını gerçekleştirmeyi amaçlamaktayız. Bir çok gerçek zamanlı nesne tanıma çalışması literatürde önerilmiş olmasına rağmen HOG tanımlayıcılara dayalı FPGA üzerinde donanım tabanlı nesne tanıma yaklaşımı, yazarların bilgisi dahilinde, henüz literatürde mevcut değildir.

Bu doğrultuda farklı yönelim (orientation) ve farklı ölçeğe (scale) sahip 125 farklı şeklin, HOG yöntemi ve kural tabanlı bir sınıflandırıcı kullanılarak, başarı ile tanınabildiği bir uygulama FPGA üzerinde VHDL kullanılarak gerçekleştirilmiştir.

Bu makale şu şekilde organize edilmiştir. Bölüm 2’de HOG yöntemi kısaca tanıtılmıştır. Bölüm 3 HOG yönteminin FPGA uyarlanması için önerilen yaklaşım verilmiştir. Elde edilen sonuçlar bölüm 4’de sunulmuştur.

2. HOG

Son yıllarda imge tanıma konsepti içinde kenar oryantasyon histogramlarının kullanımı büyük bir ivme kazanmıştır. Ancak yoğun ve lokal gradyan oryantasyon histogramlarının (HOG) kullanımı ilk defa Shashua [1] ve Dalal [2] tarafından önerilmiştir. Bu yöntemdeki amaç imgeyi bir grup lokal histogramlar olarak tanımlamaktır. Bu histogramlar, imgenin lokal bir bölgesindeki gradyanların oryantasyonlarının oluşma sayılarını içerir. HOG’ tanımlayıcıları gerçekleştirilmesindeki işlem basamakları şu şekilde sıralanabilir:

- 1) İmgenin gradyanının alınması,
- 2) Belirlenen lokasyonlar için oryantasyon histogramlarının oluşturulması
- 3) Oluşturulan lokasyon grupları içindeki histogramların normalize edilmesi.

Ayrıca performansı geliştirmek için imgede blok adı verilen büyük bir bölge arasında yoğunluk ölçümü hesabı yapılarak yerel histogramlar normalize edilir.

2.1 Gradyan Hesaplama

İmgenin gradyanı imgenin iki boyutlu filtreler kullanılarak süzgeçlenmesi ile elde edilir. Bir çok yöntemde, gradyan hesaplamada yatay ve dikey filtreler bir arada uygulanmaktadır. Bu yöntemin uygulanabilmesi için imge, gri imge haline dönüştürülür. Gri imge belirlenen filtre çekirdeği ile evrişim (konvolüsyon) işlemine tabi tutulur. Bu çalışmada filtre olarak yatay ve dikey sobel filtreleri kullanılmıştır (Bakınız Denklem (1), Denklem (2)).

İmgenin sobel filtreleri ile konvolüsyon işlemine tabi tutulmasından sonra I_x ve I_y ile gösterilen yatay ve dikey kenarlar elde edilir.

$$I_x = I * S_y \quad (1)$$

$$I_y = I * S_d \quad (2)$$

Elde edilen imgeler kullanılarak gradyanın büyüklüğü ve yönü Denklem (3) ve Denklem (4) deki gibi bulunur.

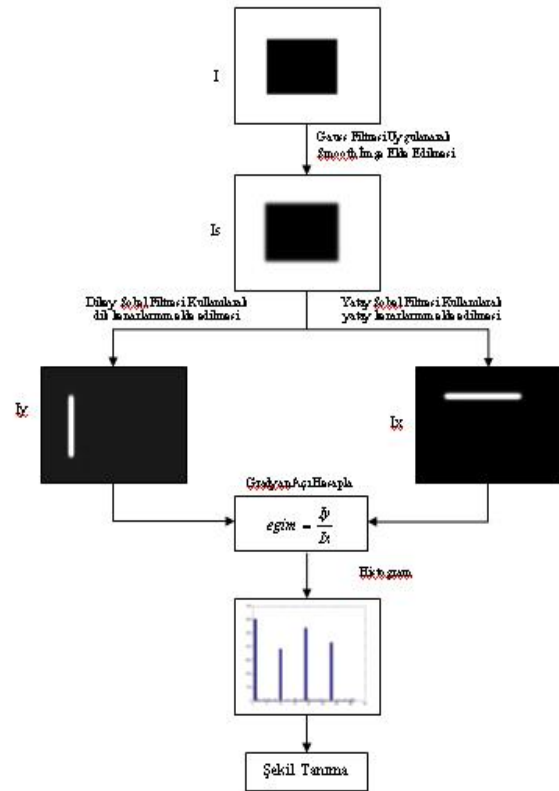
$$|G| = \sqrt{I_x^2 + I_y^2} \quad (3)$$

$$\theta = \arctan \frac{I_x}{I_y} \quad (4)$$

2.2 Yön Tablosu

Gradyanların büyüklüğü ve yön bilgileri kullanılarak lokal histogramlar elde edilir. Lokasyondaki her piksel, gradyan hesaplamada yön tabanlı histogram kanalında bulunan değer için bir ağırlık bildirir. Lokasyonlar dikdörtgenseldir ve histogram kanalları, gradyanın işaretli veya işaretli olmamasına bağlı olarak 0-180 derece veya 0-360 derece aralığına eşit olarak yayılır.

Bu çalışmada, gri imgeye sobel filtreleme uygulanmadan önce oryantasyon gürültüsünün azaltılması için düzleştirici (smoothing) Gauss filtresi uygulanmıştır. Şekil tanıma için HOG tanımlayıcılarını kullanan kural tabanlı bir sınıflandırıcı geliştirilmiştir. Tasarlanan sistem akısı (system flow) Şekil 1’de verilmiştir.



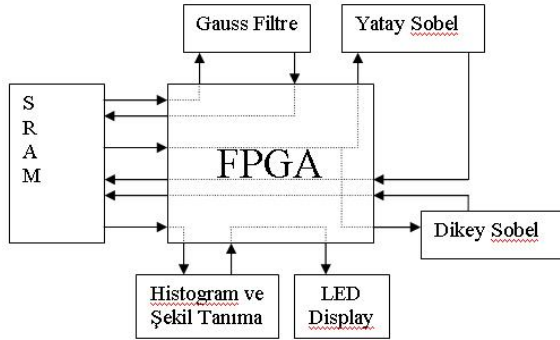
Şekil 1: Şekil tanıma işlemi blok şeması

3. FPGA Tabanlı Genlik ve Açış Hesaplanması

Test imgeleri, 8 farklı şekli ve bu şekillerin farklı yönelim ve ölçekleme ile elde edilmiş toplam 125 imgeden oluşmaktadır. Kullanılan her imge, imge büyüklüğünü standart hale getirmek için 150x112 ebatlarında ölçeklendirilmiştir ve imgeler 8 bit

derinliğe sahip olacak şekilde düzenlenmiştir. Şekil 1’de verilen akış şemasında da gösterildiği gibi, öncelikle imgelerde mevcut oryantasyon gürültüsünü bastırmak amacı ile SRAM’den okunan imgeye Gauss filtresi uygulanarak tekrar SRAM’a yazılmıştır.

Gauss filtresi uygulanan imge tekrar SRAM’dan okunduktan sonra yatay ve dikey sobel filtresi uygulanarak gradyan ve oryantasyon hesaplamasında kullanılacak olan I_x ve I_y imgeleri oluşturulmuştur. Daha sonra bu imgeler tekrar SRAM’a yazılmıştır. SRAM’a yazılan her bir imge 16.8 KByte büyüklüğünde bir yer kaplamaktadır. Yatay ve dikey kenarları elde edilen imgelerde HOG tanımlayıcılarının belirlenmesi için gerekli olan gradyan ve oryantasyon hesaplamaları yapılmıştır ve histogram tablosu elde edilmiştir. Bu histogramdan faydalanılarak önerilen kural tabanlı sınıflandırıcı yardımı ile şekil tanıma işlemi gerçekleştirilir. Yukarıda açıklanan işlem gruplarını ifade eden blok diyagram gösterimi Şekil 2’de verilmiştir.



Şekil 2: Şekil tanıma işleminin FPGA tabanlı gerçekleştirilmesi blok şeması

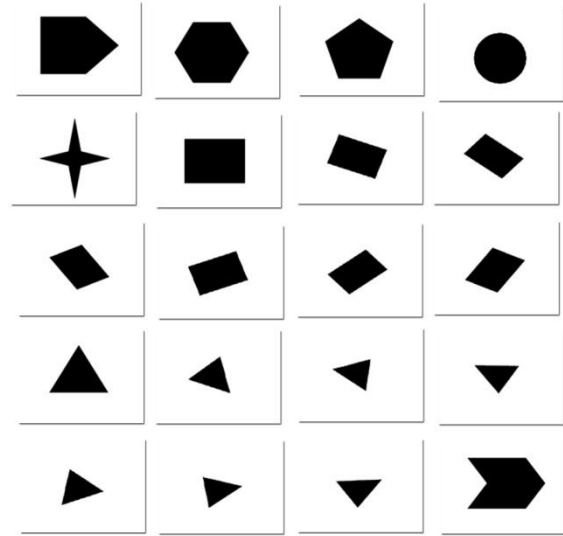
3.1. Donanım Tabanlı Histogram Oluşturma

Donanım tabanlı histogram oluşturma işlemi için Şekil 3’de verilen akış takip edilmiştir. Açık hesaplama işlemi, her piksel için Denklem (4) kullanılarak gerçekleştirilmiştir. Ancak arctanjant işlemini doğrudan donanım üzerinde uyarlanamadığından bir bakma tablosu (look-up table) kullanarak açının değeri belirlendi. Bakma tablosu 10’ar derecelik açılarla oluşturuldu. Bakma tablosu 0-90 derece arasındaki değerleri kapsamaktadır. Bu neden ile dikey kenar büyüklüğünün yatay kenar büyüklüğüne oranının mutlak değeri alınarak bakma tablosundan açı değeri belirlenir. Fakat gradyanın yönünü bulmak için ise yatay ve dikey kenar büyüklüklerinin işaretleri kullanılmaktadır. Her iki büyüklük pozitif ise açı birinci bölgede, dikey kenar büyüklüğü pozitif yatay kenar büyüklüğü negatif ise ikinci bölge, işaretler negatif ise açı üçüncü bölgede, dikey kenar büyüklüğü negatif yatay kenar büyüklüğü pozitif ise açı dördüncü bölgede yer almaktadır. Bu şekilde açılar bulunduktan sonra imgeye ait histogram elde edilmektedir.

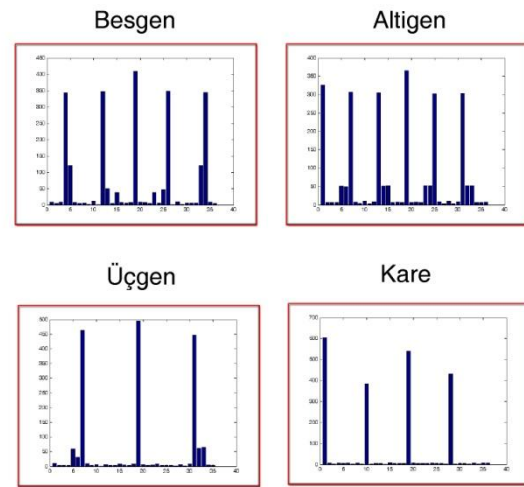
Uygulamada kullanılan bazı imgeler Şekil 4’de verilmiştir. Ayrıca Şekil 5’de, yukarıda açıklanan yöntem kullanılarak imge kümesinde mevcut olan üçgen, kare, beşgen ve altıgen şekilleri için hesaplanmış gradyan yönleri histogramları verilmiştir.



Şekil 3: Gradyan yönleri histogram oluşturma için akış şeması



Şekil 4: Veri setinde mevcut olan bazı şekiller



Şekil 5: Farklı şekillere ait gradyan yönleri histogramları

4. Sonuçlar

Literatürde son yıllarda yoğun ilgi gören HOG tanımlayıcılarına dayalı bir nesne tanıma donanımsal olarak gerçekleştirilmiştir. HOG algoritmasının yapısı paralelleştirmeye oldukça uygun olduğundan, FPGA gibi paralel işlem yapma yeteneğine sahip gömülü sistemler üzerine uyarlanmaya müsaittir.

Bu çalışmamızda bu basit fakat etkin tanımlayıcının donanımsal gerçekleştirilmesi başarı ile uygulanmıştır. Bu uygulama bir çok temel örüntü tanıma işlevinde kullanılabilirliği söz konusu olduğundan önem arz etmektedir. Çalışmamızda toplam 125 adet şekil denenmiş ve tanımda %100 lük bir başarı elde edilmiştir. Uygulama platformu olarak üzerinde Altera Cyclone II FPGA içeren Terasic DE2 bordu kullanılmıştır. Aynı zamanda bu çalışmada daha kompleks ve karmaşık ortamlarda da başarılı bir şekilde uyarlanabilecek olan bir alt yapı sunulmuştur

5. Teşekkür

Bu çalışma Sanayi ve Ticaret Bakanlığı tarafından desteklenen 0183STZ2007II nolu proje çerçevesinde gerçekleştirilmiştir

6. Kaynakça

- [1] Shashua A., Gdalyahu Y., and Hayon G.. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In Proceedings of IEEE Intelligent Vehicles Symposium, 2004.
- [2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, International Conference on Computer Vision and Pattern Recognition, volume 2, pages 886–893, June 2005.
- [3] Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded up robust features. In Leonardis,
- [4] Bellas N., Chai S. M., Dwyer M., Linzmeier D., FPGA implementation of a license plate recognition SoC using automatically generated streaming accelerators, Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, Page(s): 8 pp., 25-29 April 2006.
- [5] Smach F., Atri M., Mitéran J., Abid M., Design of a Neural Networks Classifier for Face Detection, Journal of Computer Science, vol. 2(3), pp. 257-260, 2006.
- [6] Wang F., Qin H., A FPGA based driver drowsiness detecting system, Vehicular Electronics and Safety, 2005, IEEE International Conference, pp. 358 – 363, 14-16 Oct. 2005.
- [7] Appiah K., Hunter A., A single-chip FPGA implementation of real-time adaptive background model Field-Programmable Technology, Proceedings 2005 IEEE International Conference, pp. 95 – 102, 11-14 December 2005.
- [8] Hsiao P. Y., Li L. T., Chen C. H., Chen S. W., Chen S. J., An FPGA architecture design of parameter-adaptive real-time image processing system for edge detection, Emerging

Information Technology Conference 2005, Page(s):3 pp., 15-16 Aug. 2005

[9] Ratnayake K., Amer A., An FPGA-Based Implementation of Spatio-Temporal Object Segmentation, Image Processing, 2006 IEEE International Conference, pp. 3265 – 3268, 8-11 Oct. 2006

[10] Siniscalchi S. M., Gennaro F., Vitabile S., Gentile A., Sorbello F., Efficient FPGA Implementation of a Knowledge-Based Automatic Speech Classifier, ICES 2005, pp.198-209, 2005.

[11] Wall G., Iqbal F., Liu X., Foo S., A Fast FPGA Implementation of a Unique Multi-level Tree-based Image Classifier, Florida A&M University - Florida State University.

[12] Anguita, D., Pischiutta, S., Ridella, S., Sterpi, D., Feed-Forward Support Vector Machine Without Multipliers, Neural Networks, IEEE Transactions, Vol. 17, Issue 5, pp.1328 – 1331, Sept. 2006.

[13] Anguita, D., Ghio, A., & Pischiutta, S. "A learning machine for resource-limited adaptive hardware." NASA/ESA Int. Conference on Adaptive Hardware Systems, AHS2007. Edinburgh, UK

[14] Anguita, D., Ghio, A., Pischiutta, S. & Ridella, S. "A Hardware-friendly Support Vector Machine for Embedded Automotive Applications." IEEE Int. Joint Conference on Neural Networks, IJCNN 2007. Orlando, USA

[15] Reyna R.A., Esteve D., Houzet D., Albenge M. F., Implementation of the SVM Neural Network Generalization Function for Image Processing, Fifth IEEE Int. Workshop on Computer Architectures for Machine Perception (CAMP'00), pp.147, 2000.