

Değişebilir Parça bazlı Modeller ile Kurban Tanıma

Human Victim Detection with Deformable Part Models

Bedir Yılmaz, Muhammet Balcılar, Mehmet Fatih Amasyalı, Sırma Yavuz
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
bedirylmazmail@gmail.com, {muhammet, mfatih, sirma}@ce.yildiz.edu.tr

Özetçe— Kendi kategorisi altında yeterince sabit bir şekle sahip olan nesnelerin tanınmasında başarılı olan HoG gibi algoritmalar, arama kurtarma görevlerinde karşılaşılan insan hedefler gibi, eklemli yapısı nedeniyle çok çeşitli yapıların ortaya çıktığı kategorilerde başarısız olmaktadır. Çalışmamızda bu gereksinimi karşılayan DPM in yüksek doğruluk ve hızı sahip bir versiyonunu, otonom bir robota bilgisayarla görme yeteneği kazandırılması çerçevesinde başarıyla uyguladık.

Anahtar Kelimeler — Arama Kurtarma; DPM; Kurban Tanıma.

Abstract— Algorithms that are highly accurate against object categories that have a sufficiently static shape such as HoG are incapable of recognizing the targets that have a highly articulated figures such as human beings which are the main case of search and rescue missions. In our study, we have implemented a version of DPM that reaches high accuracy and speed when it comes to find these kind of targets in digital images as a part of our goal to bring in the capability of computer vision to our autonomous search and rescue robot.

Keywords — Search and Rescue; DPM; Human Victim Detection

I. ARAMA KURTARMA

Arama Kurtarma çalışmaları, bir afete maruz kalmış yahut yaklaşan bir tehlikeyle karşı karşıya olan insanlara ulaşılmasını ve destek verilmesini amaçlayan çabaların genel adıdır [1]. Böylesi bir çaba, zorlu yaşam ve icra koşullarının yanı sıra çok sınırlı bir zaman içerisinde başarıyla gerçekleştirilmesi gereken görevler içermesi sebebiyle yüksek yetkinlikte kişi ve sistemlerce üstlenilmelidir.

II. ROBOTLARIN ARAMA KURTARMADAKİ YERİ

Arama kurtarma çalışmaları, yüksek hayati risk ve operasyonel aciliyet taşıyan doğaları gereği, otomatize bir biçimde yürütülmesinde son derece büyük faydalar taşıyan

uygulamalardan oluşur. İnsanın erişemeyeceği yahut erişmesi halinde hayati tehlikeye gireceği koşulların önemli bir kısmının robotların için aynı büyüklükte bir olumsuzluk teşkil etmeyeceği düşünüldüğünde, bu görevlerde robotların kullanımının getireceği faydalar daha belirgin bir biçimde ortaya çıkmaktadır.

Bu sebepler dolayısıyla bir yangın mahalline, bir binanın enkazına yahut bir maden göçüğüne girip içeride bulunan kurbanlarla ilgili yaşamsal veya konumsal bilgiler sunabilecek ve hatta bu afetzedelerin durumlarına uygun olan birtakım eylemler içerisine girebilecek robotların bu tür çalışmalarda yerini alabilecek olması, teknolojinin insanoğluna sunmuş olduğu fırsatların en önemlileri arasındaki yerini korumaktadır.

III. ARAMA KURTARMA VE OTONOM ROBOTLAR

Ne var ki, uzaktan kumanda edilen robotlar için göreceli olarak kolay olan bu görevlerin icrasında, yine görevlerin doğası gereği büyük bir zorlaştırıcı etmen ortaya çıkmaktadır : Erişimi zor ortamların (deprem, göçük vb.) bir çoğunda elektromanyetizmaya dayalı iletişimin de sekteye uğradığı bir gerçektir. Bu tarz bir iletişimsizlik anında robotun doğru kararlarla çalışmaya devam etmesi ise yüksek önem taşımaktadır. Hal böyleyken robotların, çevrelerinden topladıkları bilgilere dayanarak yaptıkları çıkarımlarla, otonom bir faaliyet yürütebilmeleri arama kurtarma çalışmalarında yer alabilmeleri için vazgeçilmez bir ihtiyaca dönüşmektedir.

Robotik teknolojilerinin arama kurtarma uygulamalarında istihdam edilmesi, yapay zekadan, lokalizasyona, haritalamadan, bilgisayarla görmeye birçok farklı çalışma alanına ait çözümlerin bir araya getirilmesini gerektirmektedir [2]. Bu çözüm gruplarının en önemlilerinden biri de, kameralardan alınacak sayısal görüntülerden, arama kurtarma çalışmalarının temel hedefleri olan insan tesbiti işlemlerinin bulunduğu, “İnsan Kurban Tanıma” (Human Victim Detection) dır.

IV. KURBAN TANIMA GÖREVİ İÇİN ADAY ALGORİTMALAR VE DPM

Kurban tanıma, taksonomi olarak bilgisayarla görme nin altında kalan disiplin olup, bu alanda da şekil tanıma kategorisinin bir alt başlığı olarak karşımıza çıkmaktadır. Kurban kavramı, özellikle insan vücudunun geniş açı çizebilen birçok eklemden (Highly Articulated) oluşan yapısının bir sonucu olarak ortaya çıkan şekilleri karşılamaktadır. Bu nedenle kurban tanıma sorununa birer çözüm olarak sunulan yaklaşımlar, yaya ve yüz tanıma (Pedestrian Detection, Face Detection) algoritmalarının statik ve bütüncül yaklaşımının aksine, dinamik ve bölümlayici olmak zorundadır.

HoG [3] ve Viola & Jones [4] gibi tanınmış şekil tanıma algoritmalarının bütüncül yaklaşımlı yapısı, hedef nesnelerdeki olası şekil değişikliklerinin tolare edilmesinde yeterli olmamakta, nesnelerin şekil değiştirmesi, insan vücudunda olduğu gibi, kendi kategorilerine has bir kurala binaen gerçekleşse de, bu kuralların tanımlanabileceği bir öğrenici yapısı olmaksızın, üretilen şekil tanıma modelinin fazlasıyla rijit olmasına yol açmaktadır. DPM [5] (Deformable Part Models - Değişime uğrayabilen parça bazı modeller) algoritması, böyle dinamik bir modele duyulan ihtiyaca cevaben ortaya çıkmıştır.

DPM algoritmasının özellik çıkarımında temel aldığı algoritma olan Histograms of Oriented Gradients - HoG [3] algoritması, resimlerin yoğunlaştırılmış birer örneğini çıkarılmasını işlemini piksel bazlı bir renk yoğunluğu histogramı çıkararak yapar. Bu histogramın üretilmesi işlemi için öncelikle resim 8 x 8 lik piksel hücrelerine bölünür. Daha sonra her bir hücre için, piksellerin x ve y eksenindeki türevlerinin birbirine oranlarının (dx / dy : Orientation, Açı) ve Pisagoryen toplamalarının ($\sqrt{dx^2 + dy^2}$: Magnitude, Büyüklük) yer aldığı birer boyutlu histogramlar hesaplanır.

Sonrasında daha genel bir temsil elde edebilmek adına, her bir pikselin sahip olduğu açı değerleri, bir oylama sistemi ile kendilerine yakın olan 9 açı sepetine (Orientation Bins) dağıtılmak üzere kullanılır. Renkli resimlerde her bir pikselin en yüksek büyüklük (Magnitude) değerine sahip olduğu renk kanalında sahip olduğu açığı kullanmak yaygın olarak tercih edilmektedir. Açı sepetlerinin adedi ve hücre büyüklükleri, algoritmanın uygulamaları arasında farklılık gösterebilmektedir.

Yapılan bu işlemler sonucunda elde edilen histogram, eldeki resme benzer özelliklere sahip resimleri tesbit edebilmek için bir filtre olarak kullanılır. Bu filtrenin resim üzerinde gezdirilmesine dayanan nesne tesbit yöntemlerine, kayan pencere tesbit edicileri (Sliding Window Detectors) denmektedir.

DPM algoritmasının resimleri işleme prensibi, HoG dakine benzer olarak, filtreleri temel almaktadır. Buna rağmen DPM, birçok noktada HoG dan ayrılmaktadır. Herşeyden önce DPM de nesne tesbitleri tek bir HoG ile değil, HoG piramitleri sayesinde yapılmaktadır.

Dolayısıyla tesbitlerin konumlarından ötürü kazandıkları skorlar hesaplanırken, x ve y eksenine, puanı kazandıran özelliklerin dahil olduğu HoG un piramitteki konumu olan l (level) yani seviye eksenine de katılmaktadır.

Başka bir deyişle filtreler, bir HoG piramidinin alt pencerelerinin ağırlıklarını temsil eden dikdörtgensel şablonlardır. w ye h büyüklüğündeki bir F filtresinin skor u $w \times h \times 9 \times 4$ adet ağırlıkla ifade edilir ki bu değerlerden 9, açı sepetlerinin adedine, 4 ise 2 x 2 lik hücre boyutuna karşılık gelmektedir.

H nin bir HoG piramidi, p nin de (x,y,l) şeklinde ifade edilen yatay-düşey eksenler ve piramid derinliğine bağlı konum bilgilerinden oluşan bir nokta olduğunu düşünelim. Bu durumda $\phi(H,p,w,h)$ vektörü, H nin sol üst köşesi p de olan bir alt penceresine ait HoG değerlerinin birbirine ulanmasıyla (concatenation) oluşmaktadır.

Bu yüzden bu tesbit penceresi (Detection Window) üzerindeki F filtresinin skoru $F \cdot \phi(H,p,w,h)$, veya kısaca $F \cdot \phi(H,p)$ olacaktır.

DPM algoritması, nesne tesbitinde kullanılacak filtre sayısının hedef nesne modelinin parça adediyle orantılı olmasını öngörmektedir. Tesbit işlemi, bütün tesbit penceresini kapsayan bir kök filtre (F_0) ile bu kök filtrenin HoG piramidinde birkaç seviye altında yer alan parça filtreleriyle yapılmaktadır. Bilindiği gibi piramid üzerinde aşağı inildikçe çözünürlük artmaktadır. Daha yüksek çözünürlük ise nisbeten küçük bölgeleri kapsayan özelliklerin kullanımını mümkün hale gelecektir. n parçaya sahip bir nesnenin modeli, bir F kök filtresi ve bir alt modeller kümesiyle (P_1, \dots, P_n) ile ifade edilir. P yi açıklamak için kurulabilecek bir $P_i = (F_i, v_i, s_i, a_i, b_i)$ eşitliğinde F_i , i ninci parçaya ait filtre, v_i , alt filtreyi çevreleyen kutunun (bounding box) merkezinin kök filtreye göreli olabileceği konumları içeren iki boyutlu bir vektörü, s_i , bu kutunun büyüklüğünü temsil ederken, a_i ve b_i de i inci alt parçanın skorunu hesaplamakta kullanılacak ikinci dereceden bir fonksiyonun katsayılarını taşıyan iki boyutlu birer vektörü karşılamaktadır.

Parçalı bir modelin resim üzerindeki yerleşiminin skoru (score of a placement) kök filtrenin yerleşim skoru ile alt filtrelerin kök e göreli olarak hesaplanmış yerleşimlerinin skorlarının toplamı şeklinde hesaplanmaktadır.

$$\sum_{i=0}^n F_i \cdot \phi(h, p_i) + \sum_{i=1}^n \alpha_i \cdot (\tilde{x}_i, \tilde{y}_i) + b_i \cdot (\tilde{x}_i^2, \tilde{y}_i^2) \quad (1)$$

Özellik çıkarımında HoG u temel alan ve genişleten bu algoritma, eğitilen yapay öğreniciler açısından bakıldığında da HoG dan farklılaşmaktadır. SVM [6] ler yerine Latent SVM [7] adı verilen ve tesbit edilecek şablonun yanı sıra, bu şablonun alt parçalarının aralarında var olması muhtemel ilişkilerin de gizli birer değişken olarak tutulduğu öğrenicilerin kullanılması, DPM in kullanıma

alanını genişletmiş ve nesne tesbitlerindeki başarı oranını önemli bir oranda artırmıştır.

DPM algoritmasının Ross Girschick tarafından düzenlenilen dağıtımı olan voc-5 [8] uygulaması üzerinde yaptığımız testler, algoritmanın kullanımının çalışma şartlarımıza uygun hale getirilmesi işleminde önemli bir basamak olmuştur.

DPM in hedef nesnelerin şekil değiştirmelerine gösterdiği toleransın yüksekliği, algoritmayı arama kurtarma faaliyetleri dahilinde çalışacak bir kurban tanıma uygulaması için iyi bir aday yapmaktadır fakat aynı işlemin başka bir boyutu vardır ki o da DPM in, HoG [3], Cascaded Features [4], SIFT [9], gibi algoritmaların eğittiği sıradan SVM lerle karşılaştırıldığında çok daha yavaş kalışıdır. DPM in çalışma hızı açısından yetersizliği, özellikle kurban tanıma misyonu göz önünde bulundurulduğunda büyük bir problem olarak ortaya çıkmaktadır.

V. DPM VE KURBAN TANIMA

Kurban tanıma uygulamalarında hızın çok kritik oluşunun en temel nedeni, bu uygulamaların otonom bir robotun üzerinde koşacak olmasıdır. Arama kurtarma çalışmalarında kullanılan bir otonom robot, kurban tanıma uygulamasından dönecek değerlere göre hedef saptaması yapıp bu hedefe yöneleceğinden, kameradan gelen verinin robotun durmasına ihtiyaç doğurmayacak bir sıklıkla taranması gerekmektedir. Bu sıklık, zamana karşı yarışan otonom robotun istenen cevabı hızlıca alıp karar alma mekanizmasına iletebilmesi için çok önemlidir.

Çalışmamızda, doğruluk açısından tatmin edici değerler üretebilen bir algoritma olan DPM e ulaştıktan sonra, bu mekanizmanın daha hızlı çalışabilmesi için çeşitli alternatifleri değerlendirmeye başladık. DPM algoritmasının Pedro Felzenszwalb ve ekibi tarafından yayınlanan [10] versiyonunun, gerçek zamanlı uygulamalar için tercih edilesi olmayan bir yaklaşımla ve MATLAB dilinde yazılmış olması bizi, araştırmalarımızı görüntü işleme çalışmalarında yaygın olarak kullanılan C++ dili üzerinde yoğunlaşmaya yöneltti.

2.9 GHz CPU ya sahip bir bilgisayar üzerinde 640 x 480 piksellik bir resmi yaklaşık 2.5 saniye sürede tarayan MATLAB kodunun yerine kullanılacak yeni programın, aynı görevi bir saniyenin altında bir süreyle yapabilmesi seçim kriterlerimiz arasındaki gerek şartlardan biriydi. Araştırmalarımızın ilk ayağında OpenCV kütüphanesi bünyesindeki “Latent SVM” [7] adındaki API nin, DPM in OpenCV üzerindeki uygulaması olduğunu gördük. Fakat bu gerçeklemenin de, benzer koşullarda (2.6 GHz CPU) tek thread ile koştuğunda aynı büyüklüklerdeki bir resmi 2 saniyenin altında dahi işleyemiyor oluşu, bizi arayışımızı sürdürmeye yöneltti. Bunun ardından ortaya seçenekler arasında libCCV [10] gibi bu süreyi 1.7 saniyeye düşürdüğü belirtilen ve FFLD [11] gibi aynı görevi 0.2 saniyede gerçekleştirebilecek kadar hızlı algoritmalar mevcuttu. DPM [5] in doğruluk açısından sağladığı

kazançtan ödün vermemeyi önceleyerek yaptığımız testler, bize FFLD nin, Charles Dubout’un ilgili makalesindeki hıza erişebildiğini, dolayısıyla sistemimizde kullanımının faydalı olacağını gösterdi ve çalışmamızda yeni bir döneme girmiş olduk.

FFLD algoritmasının sabit INRIA [12] veri kümesi üzerinde göstermiş olduğu başarının ardından, OpenCV kütüphanesi yardımıyla derlediğimiz kodun makul bir düzeyde (Release optimizasyonları ve paralel çalışma olmaksızın ~1 resim/sn) performans sergilemesiyle birlikte, bu çalışmanın robotumuzun ROS (Robot Operating System) [13] üzerine taşınması için gereken hazırlık basamakları tamamlanmış oldu.

ROS un çalışmamızdaki önemi, kurban tanıma uygulamamızın üzerinde koşacağı robota verilen komutların tamamının bu sistem üzerinden yönetiliyor olmasından ileri gelmektedir. Python ve C++ dillerinde geliştirilen kodların, küçük değişikliklerle, ROS üzerinde çalışan uygulamalara has ROS “Düğüm” (Node) lerine dönüştürülebilmesi mümkün olmaktadır. Yine bu işletim sistemi üzerinden OpenCV kütüphanesine erişimine olanak sağlayan `ros::cvBridge` adı verilen API nin de yardımıyla, OpenCV uygulamamızdaki webcam in yerini robotumuza bağlı olan Kinect’e bırakmasıyla kurban tanıma uygulamamızın, 7 fps ile çalışan kurban tanıma yeteneğini robotumuza kazandırmış olduk.

FFLD algoritmasının yüksek derecedeki hızı, doğruluk açısından da DPM den geri kalmayan bu algoritmayı uygulamamıza katmamız kararında etkili oldu. HoG algoritması ile nesne tesbitinde bir filtrenin skoru, resim üzerinde eşleştiği piksellerin özellikleriyle kendi özelliklerinin uyumlarına bağlı olarak elde edilmektedir.

K nin toplam özellik sayısını, x in $M \times N$ lik bir resmi ve y nin $P \times Q$ luk bir filtreyi temsil ettiği bir denklemde z nin de bahsi geçen filtrenin resme uygulandığı sonucu ortaya çıkan skorlara karşılık geldiğini düşünelim. Bu durumda bütün kayan pencere taramalarının adedinin denklemi (2) deki gibi olur.

$$z_{ij} = \sum_{k=0}^{K-1} \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} x_{i+p,j+q}^k y_{pq}^k \quad (2)$$

Charles Dubout ve diğerleri [11], bir lineer filtrenin resmin bütün noktalarıyla ayrı ayrı beslendiğinde ortaya çıkaracağı cevapları, resim ile lineer filtrenin tersinin konvolüsyonlarının toplamıyla da bulunabileceğini kaydetmişler ki bu da yukarıda belirttiğimiz işlemi

$$z = \sum_{k=0}^{K-1} x^k \bar{y}^k \quad (3)$$

Şeklinde kısaltabilmemiz anlamına gelmektedir. Böylesi bir algoritma, zaman maliyeti açısından değerlendirildiğinde ortaya, $M \times N$ lik bir resim

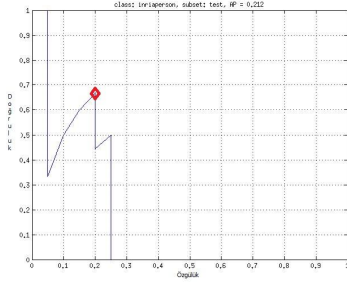
ve $P \times Q$ luk bir filtre için $O(MNPQ)$ luk en kötü durum maliyeti çıkar.

Bu maliyet, bir de K adet özellik ve L adet filtre için düşünüldüğünde ortaya çıkacak değer $O(KLMNPQ)$ kadar büyük olacaktır. Fourier transformasyonunun sağladığı katkıyla konvolüsyon maliyeti $O(MN \log MN)$ olmakta ve filtreler bu işlemle hiçbir doğruluk yahut hassasiyet kaybına uğramamaktadır.

VI. TESTLER

Robocup Rescue Leage [15] adı verilen, her yıl uluslararası olarak düzenlenen ve robot teknolojilerini konu alan yarışmaların standartlarına uygun olarak düzenlenmiş laboratuvarımızda elde ettiğimiz test ve eğitim verilerini, voc -5, fflvd ve fflvd2 uygulamalarının eğitilmesinde ve testlerinde kullandık. Kullandığımız eğitim ve test verilerinin tamamı 640x480 piksellik üç kanallı RGB resimlerden oluşmaktadır. Eğitim verilerimizin kümesi 142, test verilerimizin kümesi ise 134 ü pozitif etiketli olmak üzere toplam 270 resimden oluşmaktadır.

Uygulamamızın istenilen düzeye gelebilmesi için yaptığımız bir başka geliştirme, DPM in çalıştırılmasında kendi problem alanımıza uygun bir eşik değerinin saptanması olmuştur.



Grafik 1

Eşik değerinin algoritmadaki karşılığı, tesbit edilen nesneler arasından hangilerinin olumlu doğru tesbitler olarak sınıflandırılacağına dair bir skor yani z değeri sınırıdır. Bu sınır, Grafik 1 deki ROC eğrisinde görülen maksimum doğruluk ve minimum özgüllük esas alınarak 0,2 olarak belirlenmiştir.

Bu eşik değerinin farklı uygulamalardaki eşdeğerleriyle elde ettiğimiz sonuçlar Tablo 2 de verilmiştir :

| Uygulama Adı / Ölçüt | Doğru Olumlu Tesbitler | Doğru Olumsuz Tesbitler | Süre (Çok İşlemci) | Süre (Tek İşlemci) |
|----------------------|------------------------|-------------------------|--------------------|--------------------|
| voc - 5 | 41/134 (%30.5) | 136/136 | 2.461 sn | 8.743 sn |
| ffldv2 | 53/134 (%39.5) | 136/136 | 0.204 sn | 0.254 sn |

Tablo 1

VII. SONUÇ

Yapılan testler ve uygulamalar göstermiştir ki, DPM algoritması, FFLD nin getirdiği yenilikler ve doğru eşik değerleriyle çalıştırıldığında arama kurtarma görevleri için uygun bilgisayarla görme metodu haline gelmektedir. Görece düşük olarak nitelendirilebilecek %39.5 lik doğru saptama oranına rağmen bu algoritma, saniyede 7 tarama gerçekleştirebildiğinden ve hatalı tesbitlerin ortaya çıkmaması konusundaki güvenilirliğinden dolayı otonom bir robot üzerinde kullanılmak için yeterince başarılıdır, zira otonom bir robotu istenilen verimle çalışmasının önündeki asıl engel kurbanın resimlerin ilk taranmasında (0,07 saniye içinde) bulunamaması değil, hatalı bir şekilde insan bulunduğunu zannedip sehven eylemde bulunmasıdır.

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu TÜBİTAK (EEEAG- 113E212) tarafından desteklenmiştir.

KAYNAKÇA

- [1] Canadian Forces (May 1998). "B-GA-209-001/FP-001 DFO 5449 NATIONAL SAR MANUAL" (PDF). Archived from the original on 2008-08-03. Retrieved 2008-07-12.
- [2] S. Bahadori, D. Calisi, A. Censi, A. Farinelli, G. Grisetti, L. Iocchi, and D. Nardi. (2005) Autonomous systems for search and rescue. In A. Birk, S. Carpin, D. Nardi, Jacoff A., and S. Tadokoro, eds. Rescue Robotics. Springer-Verlag
- [3] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005
- [4] Viola, Paul, Michael J. Jones, and Daniel Snow. "Detecting pedestrians using patterns of motion and appearance." Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003
- [5] Felzenszwalb, Pedro, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model." Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008
- [6] Hearst, Marti A., et al. "Support vector machines." Intelligent Systems and their Applications, IEEE 13.4 (1998)
- [7] http://docs.opencv.org/modules/objdetect/doc/latent_svm.htm
- [8] <http://www.cs.berkeley.edu/~rbg/latent/>
- [9] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.
- [10] <http://libccv.org/doc/doc-dpm/>
- [11] C. Dubout and F. Fleuret. Deformable Part Models with Individual Part Scaling. In Proceedings of the British Machine Vision Conference (BMVC), 2013
- [12] Navneet Dalal, <http://pascal.inrialpes.fr/data/human/>
- [13] <http://www.ros.org/>
- [14] <http://www.robocuprescue.org/>