

## GÖRÜNTÜ İŞLEMEDE NESNE KOORDİNAT ÖZELLİKLERİNİ KULLANARAK BAKLIYAT SAYMA İŞLEMİNE BİR YAKLAŞIM

<sup>1</sup>Muhammet Üsâme ÖZİÇ , <sup>2</sup>Nihat ÇANKAYA , <sup>3</sup>Muciz ÖZCAN , <sup>4</sup>Barış GÖKÇE 

<sup>1</sup>Necmettin Erbakan Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Biyomedikal Mühendisliği Bölümü, Konya, TÜRKİYE

<sup>2</sup>Necmettin Erbakan, Meram Meslek Yüksekokulu, Gıda İşleme Bölümü, Konya, TÜRKİYE

<sup>3</sup>Necmettin Erbakan Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Konya, TÜRKİYE

<sup>4</sup> Necmettin Erbakan Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Mekatronik Mühendisliği Bölümü, Konya, TÜRKİYE

<sup>1</sup> [muozic@erbakan.edu.tr](mailto:muozic@erbakan.edu.tr), <sup>2</sup> [ncankaya@erbakan.edu.tr](mailto:ncankaya@erbakan.edu.tr), <sup>3</sup> [mozcan@erbakan.edu.tr](mailto:mozcan@erbakan.edu.tr), <sup>4</sup> [bgokce@erbakan.edu.tr](mailto:bgokce@erbakan.edu.tr)

(Geliş/Received: 06.11.2020; Kabul/Accepted in Revised Form: 03.12.2020)

**ÖZ:** Nesne sayma, gıda, medikal, endüstri ve günlük yaşamda farklı görevler için kullanılan bir süreçtir. Bu çalışmada, nesne sayma işleminin gerçekleştirilebilmesi için görüntü işleme tabanlı sistemler incelenmiş ve uygulamalar yapılmıştır. Uygulamalar için nesnelerin bir silo üzerinden akararak serici üzerinden eğimli bir şekilde ilerleyebileceği bir deney düzeneği tasarlanmıştır. Nesnelerin akışını gözlemleyebilecek endüstriyel bir kamera ve lens sistemi kullanılmıştır. Nesnelerin akışını izleyebilmek için sadece arka aydınlatma kullanılmıştır. Nesnelerin akış hızı ve nesnelerin serici çıkışı akabileceği eğim açısı değiştirilebilmektedir. Endüstriyel kamerada fps, piksel frekansı, pozlama süresi, görüntünün çözünürlüğü, ilgilenen alan seçimi, renkli ve renksiz görüntü alımı, görüntüdeki piksellerin kaç bit ile örnekleneceği kullanıcı tarafından seçilebilmektedir. Algoritma tasarımı için Python yazılımı ve OpenCV kütüphanesi kullanılmıştır. Nesne sayımı için 100 adet nohut tanesi belirlenmiş ve belirli bir süre kameradan video kaydı elde edilmiştir. Video üzerinde arka plan çıkarma işlemi uygulanarak sadece nesnelerin beyaz renk olarak görülebileceği ikinci bir video elde edilmiştir. Binary formata dönüştürülen videoda nesne çevresi sınırlayıcı en küçük dikdörtgen koordinat değerleri ve nesne ağırlık merkezi koordinat değerleri elde edilmiştir. Video görüntüsü üzerinde sabit tek ve çift sanal çizgiler çekilerek sayma işleminin gerçekleştirileceği metotlar geliştirilmiştir. Koordinat değerlerinin ve sanal çizgilerin sayma işlemi için kullanılmasında ortaya çıkan avantaj ve dezavantajlar bu çalışma sonucunda tartışılmıştır.

**Anahtar Kelimeler:** Nesne Sayma, Görüntü İşleme, Kamera, Python, OpenCV

### An Approach to Counting Legumes Using Coordinate Features in Image Processing

**ABSTRACT:** Object counting is a process used for different tasks in food, medical, industry and daily life. In this study, image processing-based systems have been examined and applications have been made to perform the object counting process. For the applications, an experimental setup has been designed in which the objects can flow over a silo and proceed in an inclined manner over the spreader. An industrial camera and lens system that can observe the flow of objects were used. Only backlighting was used to monitor the flow of objects. The flow rate of objects and the angle of inclination through which objects can flow out of the spreader can be changed. In the industrial camera, fps, pixel frequency, exposure time, image resolution, area of interest selection, color and mono image acquisition, how many bits the pixels in the image will be sampled can be selected by the user. Python software and OpenCV library were used for

algorithm design. For the object counting, 100 chickpea seeds were determined and the video was recorded from the camera for a certain period of time. By applying a background subtraction process on the video, a second video was obtained in which only the objects can be seen as white. The smallest rectangular coordinate values bounding the object and the object centroid coordinate values were obtained in the video converted into binary format. Methods have been developed for counting by drawing single and double virtual lines in the video. The advantages and disadvantages of using coordinate values and virtual lines for counting have been discussed as a result of this study.

**Key Words:** *Object Counting, Image Processing, Camera, Python, OpenCV*

## GİRİŞ (INTRODUCTION)

Görüntü sensörleri ile nesne sayma süreçleri, endüstri, gıda, medikal, güvenlik gibi geniş bir sektörde kullanımına ihtiyaç duyulan teknolojilerdir. Nesne sayısının küçük ve çok fazla olması, nesnelerin aynı anda hareketli olması manuel bir şekilde insan gücüne bağlı olan sayma işlemlerini oldukça zorlaştırmaktadır. İnsan gözünün bir video görüntüsünden çoklu nesne takibini kaçırabilmesi, bir bant üzerinden veya serbest düşme ile akan nesnelerin hızını yakalayamaması sayma işlemini zorlaştırmakta, süre problemini ortaya çıkarmakta ve hata yapmaya açık süreçlerin ortaya çıkmasına neden olmaktadır. Bundan dolayı görüntü sensörlerini kullanan endüstriyel kamera teknolojilerinin yapay görü, makine öğrenmesi, yapay zekâ, görüntü işleme ve gerçek zamanlı hızlı işlemler yapabilen donanımlar ile beraber entegre bir şekilde kullanılmasına dönük birleşik sistemler son zamanlarda endüstride talep gören teknolojilerdir. Görüntü sensörlerinin ucuzlaması ve görüntü işleme cihazlarının makul düzeylerde olması, ürün sayma çözümlerini kolay ulaşılabilir hale getirmiştir. Hazır olarak kullanılan var-yok sensörler yerine daha karmaşık nesnelerin veya ürünlerin sayılması için görüntü sensörleri tercih edilmektedir. Sayım ve ölçüm için uzman kontrolünün gerektiği durumlarda tamamen temassız çözümler geliştirilebilmektedir. Otomobil yedek parçaları, çuvallar, tohumlar, ilaçlar, bitkiler, meyveler, sebzeler, hırdavat parçaları, küçük boyutlu oyuncaklar gibi ticari değeri olan birçok endüstriyel ürün için özel çözümler gerçekleştirilebilmektedir. Sadece endüstriyel ürünler değil aynı zamanda özel kurum ve kuruluşların giriş çıkışlarının denetimi, kamu kurumlarının giriş çıkışlarının denetimi, mağazalar ve alışveriş merkezleri, güvenlik amaçlı sayma, trafik yoğunluğu, otopark doluluğu gibi sivil hayatta karşılık bulan kullanımları giderek artmaktadır (Barbedo, 2012; Raman ve Sukanya, 2012; Perera ve diğ., 2013; Thammasorn ve diğ., 2013; Pandit ve diğ., 2014; Baygin ve diğ., 2018). Alan taramalı kameralar ile yapılan sayma işlemlerinde, nesnenin kadrāja girip çıktığı ana kadar kendine ait benzersiz bir kimliğe sahip olması gerekmektedir. Videoda nesne sürekli hareket halinde olduğu için hız, birbirine yapışık veya üst üste binmiş nesnelere, kameranın çekim parametre ayarları, aydınlatma yöntemi gibi birçok faktör nesne sayma işlemini zorlaştırmaktadır. Bundan dolayı nesneye özel uygulama düzeneğinin geliştirilmesi görüntü işleme algoritma performansını etkileyecek en önemli etkidir. Nesne sayma metotları nesne algılama ve sayma, yörünge kümeleme ile sayma, global regresyon ile sayma ve nesne yoğunluk tahminine dayalı sayma olmak üzere genel olarak dört başlık altında toplanmaktadır. Nesne algılama ve temel eşik kuralına göre sayma en sık kullanılan yöntemlerden biridir. Bu yöntemin en önemli avantajı çok hızlı sayma gerçekleştirmesidir, dezavantajı ise birden fazla nesnenin birbirine yaklaşması durumunda sayım performansının düşmesidir (Lin ve diğ., 2010; Wang ve Wang, 2011; Arteta ve diğ., 2012). Yörünge kümeleme tekniği kullanılarak yapılan sayma işleminde ise kalabalık olan hareketli nesnelere sayılır (Antonini ve diğ., 2006; Rabaud ve Belongie, 2006). Bu yöntem yalnızca istenilen yörüngeler doğrultusunda nesnelere izlemek için uygulanır. Bu yöntemin en önemli dezavantajlarından bir tanesi ise yüksek hesaplama maliyetine ihtiyaç duymasıdır. Bu sebeple bu yöntemin düşük maliyetli bir çözüm olarak sunulması mümkün değildir. Ayrıca global regresyon yöntemi ile yapılan sayma işleminde hızlı eğitim ve test prosedürleri kullanıldığından tahminlerde iyi bir sonuç vermektedir. Bununla birlikte tüm görüntü boyunca tahmin yapılacak nesne sayısı, yoğunluk haritasının integrali olarak hesaplanır. Bu yöntem diğer yöntemlerden farklı olarak görüntü boyunca nesne dağılımının analizi

için nesnenin yoğunluk haritasını verir (Chan ve diğ., 2008; Ryan ve diğ., 2009; Chan ve Vasconcelos, 2011; Chen ve diğ., 2013). Son dönemlerde popüler yöntemlerden biri olan derin öğrenme yöntemleri de sayma işlemleri için kullanılmaya başlanmıştır. Ancak derin öğrenme yöntemlerinin hesapsal maliyetleri oldukça fazladır (Zhang ve diğ., 2015; Onoro-Rubio ve López-Sastre, 2016; Zhang ve diğ., 2016). Görüntü işlemede nesne sayma için derin öğrenme yaklaşımı aslında tek yönlü bir regresyondur. Bu ağlar, olası çıktı birimlerinin sayısının eğitim verilerine dayalı olarak önceden tanımlanmış bir sınıflandırma mimarisini benimser (Öztürk ve Özkaya, 2020). Bunun yanında sayılacak nesnenin toplam sayısına mümkün olduğunca yakın gerçek bir değer üreten tek bir çıktı birimine sahiptir (Aich ve Stavness, 2017; Aich ve diğ., 2018). Her iki durum da yüksek boyutlu doğrusal olmayan regresyon modelleri kategorisi altında sınıflandırılabilir. Burada dikkat edilmesi gereken tahmin edicilerin sayısı giriş piksellerinin sayısı ile orantılıdır. Son dönemlerde derin öğrenme kütüphanelerinin versiyon güncellemeleri ile beraber Tensorflow ve YOLO kütüphanelerinin nesne sayma API'leri (Application Programming Interface) pratik kullanımlar sunmaktadır (Mazurek, 2020; Özlü, 2020). Bu çalışmada, tasarlanan bir deney düzeneği ile sabit bir bant üzerinde ilerleyen 100 adet nohut tanesinin videosu elde edilmiş ve nesnelerin sayma işlemi yapılmıştır. Uygun video çekimi için kamera parametreleri optimum şekilde ayarlanmış ve nesne arkasından aydınlatma işlemi yapılmıştır. Video görüntülerinde arka plan çıkarma işlemi yapılarak nohut taneleri ön plana çıkarılmıştır. Binary formata dönüştürülen videoda nesnelerin çevresini sınırlayan en küçük dikdörtgenin koordinat değerleri ve nesnenin ağırlık merkezi koordinat değerleri elde edilmiştir. Video görüntüsü üzerinde çizilen tek ve iki sanal çizgi referansı ile nesne koordinat değerlerinin çizilen sanal çizgilerden geçip geçmediği tespit edilerek sayma işlemi gerçekleştirilmiştir. Kullanılan yöntemlerin sayma problemi için avantaj ve dezavantajları tartışılmıştır.

## MATERYAL VE METOT (MATERIAL AND METHOD)

### Deney Düzeneği (Experimental Setup)

Hareketli nesne görüntüsünün alınması için, nesnelerin silo içerisinden serici üzerine düştüğü ve sabit eğimli bir bant üzerinde ilerlediği bir düzenek tasarlanmıştır. Serici hızı ve bantın eğimi kullanıcı tarafından ayarlanabilmektedir. Deney düzeneğinde IDS marka (UI-3290SE-HQ) renkli bir endüstriyel kamera ve AZURE marka bir lens sistemi (35mm) kullanılmıştır. Aynı zamanda gelecek çalışmalar için küçük nesne ölçümlerinin yapılabileceği telesentrik bir lens temin edilmiştir. Bu kameranın çalışma için seçilme nedeni, üreticinin bu modeller için Python desteği sunması ve gelecek çalışmalarda GPU denemeleri için NVIDIA Jetson kartları desteklemesidir. IDS firması kamera kullanımı için bir arayüz (cockpit) sunmakta ve kameranın birçok parametresi buradan değiştirilebilmektedir. Çözünürlük, fps, piksel frekansı, pozlama süresi, kazanç, gama faktörü, gain faktörü, renk örnekleme bit seviyesi gibi video çekimini etkileyecek birçok parametre değiştirilebilmektedir. Bu parametreler üretici firmanın sunmuş olduğu Python kütüphanesi fonksiyonları ile gerçek zamanlı geliştirilebilecek yazılım içerisinden de kontrol edilebilmektedir. Nesnelerin sadece tane sayısı hesaplandığı, renk veya çürük gibi özellikler ile ilgilenilmediği için arka plan aydınlatma kullanılmıştır. Düzeneğin etrafı siyah bir naylon ile kaplanarak diğer ışık kaynaklarından etkilenmesi engellenmiştir. Şekil 1a'da deney düzeneği, Şekil 1b'de arka plan aydınlatma, Şekil 1c'de kamera ve lens çeşitleri verilmiştir.



Şekil 1. (a) Deney Düzenegi (b) Arka plan aydinlatma (c) Kamera ve lens çeşitleri

Figure 1. (a) Experimental Setup (b) Background illumination (c) Camera and lens types

Nesneler sericiden gelerek sabit eğimli bir bant üzerinden aktıkları için hızlı ilerlemektedir. Bundan dolayı pozlama süresi (exposure time) olabildiğince kısa tutulmuştur. Diğer parametreler kamera üretici firmanın sunmuş olduğu arayüz üzerinden manuel bir şekilde ayarlanmış ve yine arayüz üzerinden kullanıcıya sunulmuş olan video kayıt sistemi ile kayıt işlemi yapılmıştır. Çizelge 1’de çalışma için kullanılan kamera parametreleri verilmiştir.

Çizelge 1. Video kaydı için kamera parametreleri

Table 1. Camera parameters for video recording

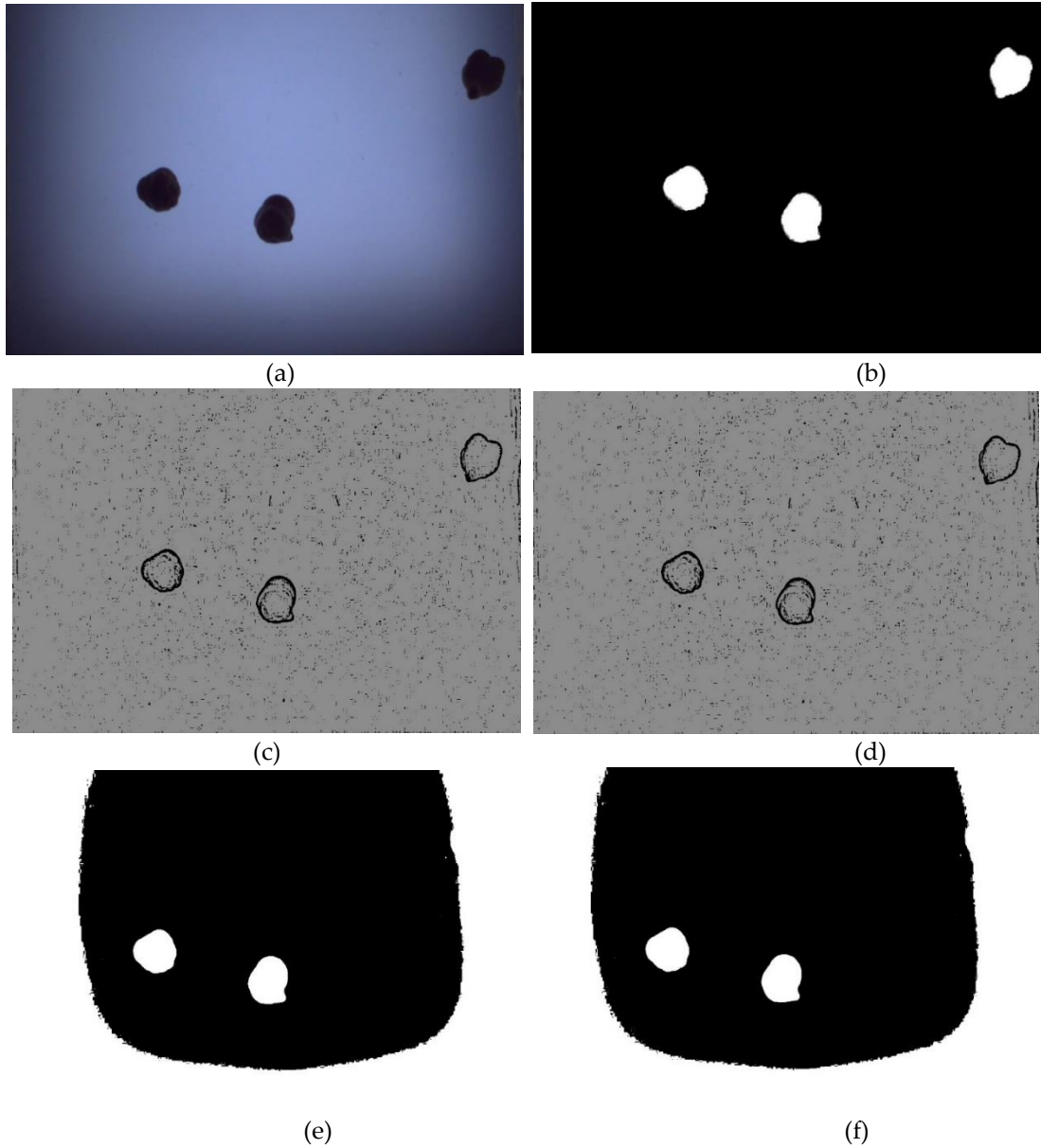
Kamera Parametreleri	
FPS	46.14 kare/saniye
Pozlama süresi	1.410 ms
Piksel Frekansı (Pixel Clock)	395 MHz
Gain Faktörü	20
Gamma Faktörü	1.6
Renk Seviyesi	RGB32 (B8 G8 R8)
Toplam görüntü kare sayısı	2399
Görüntü boyutu	1144x774 (binning 2x)

## GÖRÜNTÜ İŞLEME (IMAGE PROCESSING)

### Segmentasyon (Segmentation)

Videoda ilerleyen hareketli nesnelerin koordinat özelliklerinin bulunabilmesi için nesnelerin arka plandan ayrılarak binary formata dönüştürülmesi gerekmektedir. Bu işlem için çalışmada basit bir eşik değeri, otsu eşikleme metodu, adaptif gaussian eşikleme (adaptive gaussian thresholding) ve adaptif ortalamalar eşikleme (adaptive means thresholding) algoritmaları denenmiş ancak yeterli performans alınamamıştır (OpenCV, 2020a). Özellikle basit ve otsu eşikleme yöntemlerinde deney düzenegindeki arka plan aydinlatma platformunun ışık homojensizliğinden kaynaklı olarak yetersiz segmentasyon işlemi ortaya çıkmıştır. Adaptif gaussian eşikleme ve adaptif ortalamalar eşikleme algoritmaları ışık homojensizliğinden diğer eşikleme yöntemleri kadar etkilenmese de görüntüde çok fazla gürültü ortaya

çıkılmaktadır. Bir diğer segmentasyon yöntemi ise OpenCV kütüphanesinin sunmuş olduğu video görüntülerinde hareketli nesne arka plan çıkarıcılarıdır (OpenCV, 2020b). Çalışmada gaussian karışım modelini kullanan MOG2 metodu arka plan çıkarma için kullanılmıştır. Bu yöntem kullanılarak sadece hareketli nesneler ortaya çıkarılmakta, böylece ışık homojensizliğinden kaynaklı problemlerden algoritma etkilenmemektedir. Hareketli nesne kenarlarının bazı bölgelerinde farklı gri tonlar ortaya çıktığı için basit bir eşikleme ile görüntü binary formata dönüştürülmüştür. Segmente edilmiş video görüntüsü üzerindeki ufak gürültü parçacıklarının bir nesne gibi işlem görmemesi için iki iterasyonluk erozyon işlemi uygulanmıştır. Şekil 2a'da video kaydından 1563. görüntü karesi, Şekil 2b'de MOG2 arka plan çıkarma işlemi, Şekil 2c'de adaptif ortalamalar eşikleme, Şekil 2d'de adaptif gaussian eşikleme Şekil 2e'de basit eşik değeri, Şekil 2f'de otsu eşikleme sonuçları 1563. görüntü karesi üzerinden verilmiştir.



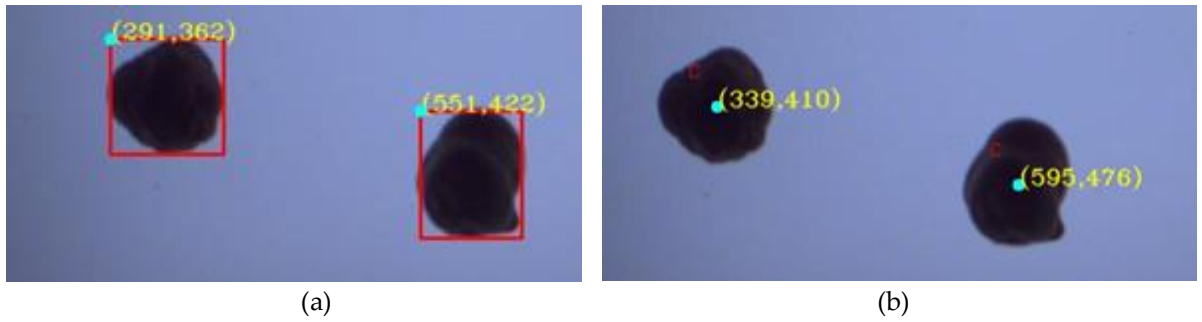
**Şekil 2.** (a) Video 1563. görüntü karesi (b) MOG2 arka plan çıkarıcı (c) Adaptif ortalamalar eşikleme (d) Adaptif gaussian eşikleme (e) Basit eşikleme (f) Otsu eşikleme

*Figure 2. (a) Video 1563. frame (b) MOG2 background subtractor (c) Adaptive means thresholding (d) Adaptive gaussian thresholding (e) Simple thresholding (f) Otsu thresholding*



### Koordinat Noktaları (Coordinate Points)

Videodaki hareketli nesnelerin ağırlık merkezi koordinatları (centroid) ve en küçük dikdörtgen içine sığabilecek kutu koordinatları (bounding box) sayma işlemi için referans olarak kullanılacaktır. Binary formata dönüştürülen görüntülerde OpenCV'nin kontur (contours) metodu ile özellikler elde edilerek ilgili koordinat noktaları elde edilmiştir (OpenCV, 2020c). OpenCV'nin dikdörtgen sınırlayıcı metodu ile (BoundingRect) nesnelerin en küçük dikdörtgen kutuya sınırlayıcı koordinatları olan  $(x, y, w, h)$  noktaları her bir görüntü karesinde tek tek bulunmuştur (OpenCV, 2020d).  $(x,y)$  noktası nesneyi sınırlayan dikdörtgenin sol üst köşesini göstermektedir. Nesnelerin ağırlık merkezi koordinatları kontur özelliklerinin momentleri bulunarak hesaplanmaktadır. Momentlerden  $m10, m00, m01$  çıktıları kullanılarak ağırlık merkezi  $(x1,y1)$  noktaları bulunmuştur (OpenCV, 2020d). Şekil 3a'da nesnelerin sınırlayıcı kutu içine alınması ve  $(x,y)$  sol üst köşe koordinat değerleri, Şekil 3b'de ağırlık merkezi ve  $(x1,y1)$  koordinat değerleri gösterilmektedir.

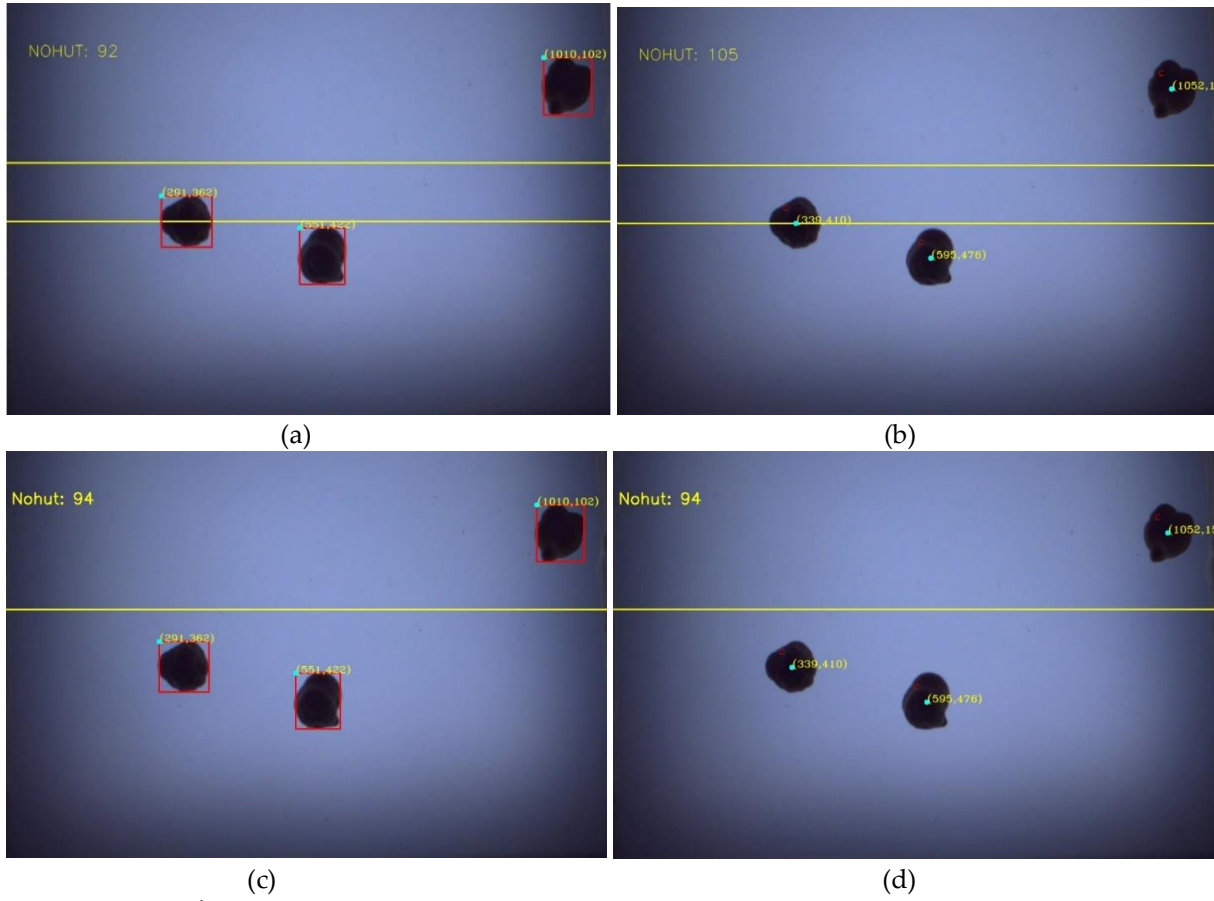


**Şekil 3.** (a) Nesnenin en küçük dikdörtgene sığdırılması ve  $(x,y)$  sol üst köşe koordinatları (b) Nesnelerin ağırlık merkezi  $(x1, y1)$  koordinat noktaları

*Figure 3. (a) Fitting the object to the smallest rectangle and  $(x, y)$  upper left corner coordinates (b) The centroid of objects  $(x1, y1)$  coordinate points*

### Sayma İşlemi (Counting Process)

Çalışmada, video görüntüleri üzerine tek ve iki sanal çizgi çizilmiş, koordinat noktaları referans alınarak geliştirilen metotlar ile sayma performansları ölçülmüştür. İlk olarak video görüntülerinin  $y=300$  ve  $y=410$  noktalarından  $x$  eksenini boyunca sanal çizgiler çizilmiştir. Ağırlık merkezi ve sınırlayıcı dikdörtgen  $y$  koordinat değeri iki çizgi arasından geçince sayaç değeri bir arttırılmıştır. İki sanal çizgi arasından geçen nesnelere ya fazla ya da az sayma işlemi yapılmaktadır. Şekil 4a'da iki sanal çizgi arasından geçen sınırlayıcı dikdörtgen koordinat değerlerinin 1563. görüntü karesinden geçişi, Şekil 4b'de iki sanal çizgi arasından geçen ağırlık merkezi koordinat değerlerinin 1563. görüntü karesinden geçişi verilmektedir. İkinci yöntem olarak  $y=300$  noktasından  $x$  eksenini boyunca tek bir sanal çizgi çekilmiştir. Tek sanal çizgi ile nesne saymada, nesnelerin çizgiye girmeden önce ve girdikten sonraki görüntü karesinden aynı nesne olup olmadığını kontrol eden bir metot uygulanmıştır. Bir görüntü karesinde bulunan nesnelerin  $x,y$  ve  $x1,y1$  koordinat değerlerinin, bir sonraki görüntü karesindeki nesnelerin  $x,y$  koordinat değerlerine uzaklıkları  $x = x(t)-x(t-1)$ ,  $y=y(t)-y(t-1)$  eşitlikleri ile hesaplanarak eksenlerdeki tüm nesne yer değiştirmeleri belirlenmektedir. Nesnenin hareketinden kaynaklı yer değiştirme uzaklığı  $x^2 + y^2$  eşitliği ile hesaplanmaktadır. Bu işlemden sonra yer değiştirme uzaklığı daha yakın olan nesne bir önceki ve bir sonraki görüntü karesindeki aynı nesne olarak işaretlenmektedir. Böylece nesne kadrardan girip çıktığı ana kadar uzaklık ölçütüne göre takip edilebilmektedir. Bu hesaplamalar yapılırken nesne koordinatının  $y=300$  sanal çizgisinden geçip geçmediği kontrol edilmekte, geçtiği anda sayaç bir defa arttırılmakta ve aynı nesne için tek bir sayım işlemi yapılmaktadır. Şekil 4c'de tek sanal çizgi arasından geçen sınırlayıcı dikdörtgen koordinat değerlerinin 1563. görüntü karesinden geçişi, Şekil 4d'de tek sanal çizgi arasından geçen ağırlık merkezi koordinat değerlerinin 1563. görüntü karesinden geçişi verilmektedir.



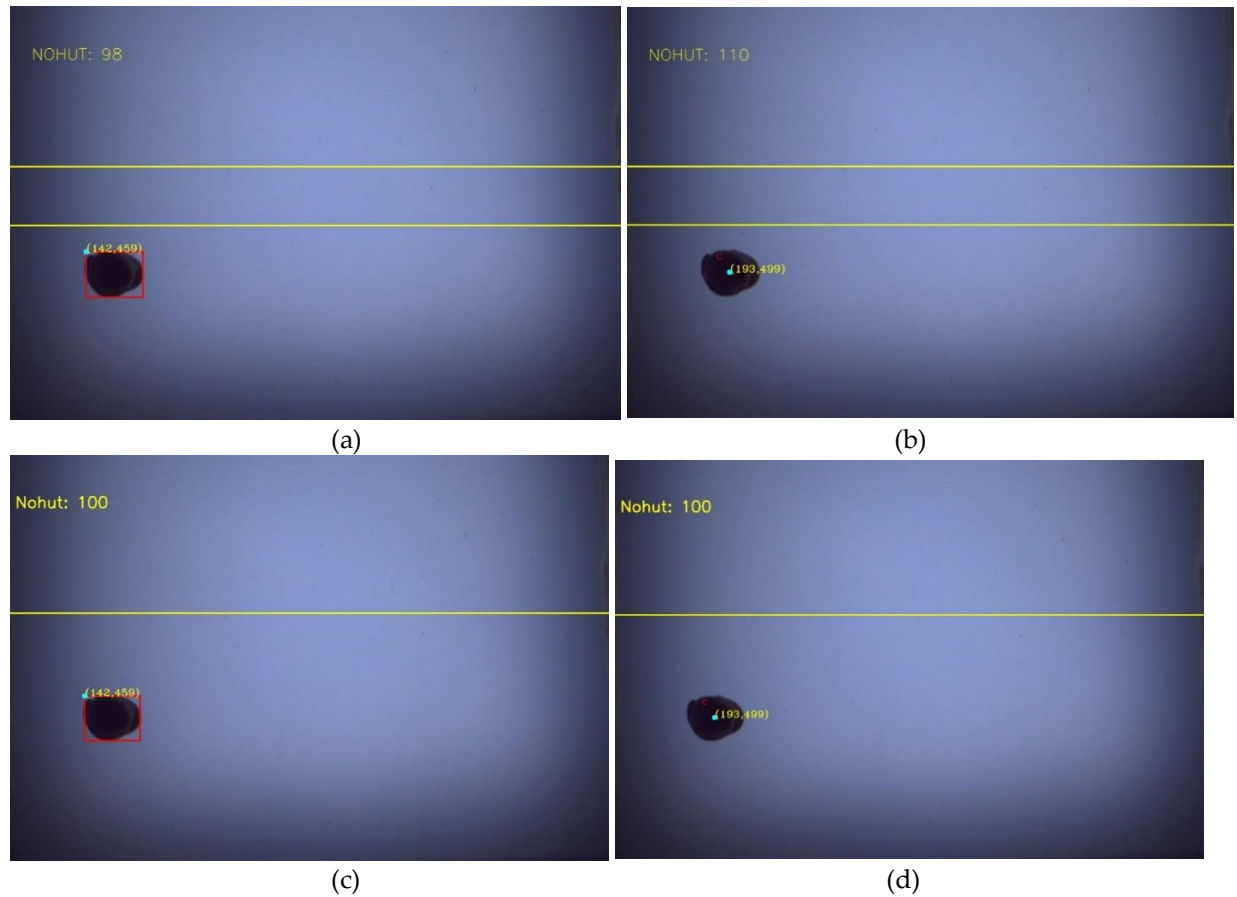
**Şekil 4.** (a) İki sanal çizgi arasından sınırlayıcı dikdörtgen koordinat değerlerinin geçmesi ve sayma işlemi (b) İki sanal çizgi arasından ağırlık merkez koordinatlarının geçmesi ve sayma işlemi (c) Tek sanal çizgi üzerinden sınırlayıcı dikdörtgen koordinat değerlerinin geçmesi ve sayma işlemi (d) Tek sanal çizgi üzerinden ağırlık merkez koordinatlarının geçmesi ve sayma işlemi

*Figure 4. (a) Bounding rectangular coordinate values passing between two virtual lines and counting process (b) Centroid coordinates passing between two virtual lines and counting process (c) Bounding rectangular coordinate values passing over single virtual line and counting process (d) Centroid coordinates passing over single virtual line and counting process*

## SONUÇ ve TARTIŞMALAR (RESULTS and DISCUSSIONS)

Bu çalışmada, kamera sistemi ve bir deney düzeneği kullanılarak kaydedilen video görüntüsünden hareketli nohut tanelerinin saydırılması işlemi gerçekleştirilmiştir. Videoda segmentasyon işlemi ile nesnelere ön plana getirilmiş ve arka plan videodan atılmıştır. Klasik segmentasyon algoritmaları ve eşikleme metotları ile başarılı sonuçlar elde edilememiştir. Bunun nedeni arka plan aydınlatmanın homojen olmamasından kaynaklı heterojen piksel intensite dağılımıdır. Videodan arka plan çıkarma işlemi OpenCV kütüphanesinin sunmuş olduğu MOG2 metodu ile gerçekleştirilmiştir. Daha sonra eşikleme işlemi yapılarak akan nesnelere binary olarak elde edildiği görüntü kareleri elde edilmiştir. Yapılan karşılaştırmalar sonucunda OpenCV'nin sunmuş arka plan çıkarıcılar videoda arka planı atmak ve istenen nesneyi ön plana getirmek için klasik görüntü segmentasyon algoritmalarına göre kullanışlı sonuçlar verdiği belirlenmiştir. Arka plan çıkarıcılar görüntü üzerindeki piksel intensite dağılımı yerine sadece görüntü kareleri arasında değişen piksel değerlerine odaklandığı için daha yüksek doğrulukta sonuçlar vermektedir. Binary formata dönüştürülmüş video üzerindeki nesnelere en küçük dikdörtgen içine sığındığı koordinatlar ve nesnenin ağırlık merkezi koordinatları OpenCV'nin moment ve kontur metodu ile elde edilmiştir. Bu koordinatlar nesne hareket ettiği sürece her bir görüntü karesinde tek tek hesaplanmaktadır. Burada temel problem nesne kadraja girdiği andan çıktığı ana kadar benzersiz bir kimliği olması gerekliliğidir. Aksi takdirde nesne her bir görüntü karesinde tekrar tekrar saydırılır. Bu

çalışmada elde edilen koordinat değerlerinin ilerleyişi iki farklı sanal çizgi yaklaşımı ile değerlendirilmiş ve sayma performansları ölçülmüştür. İki sanal çizgi arasından ilgili nesne koordinatları geçtiğinde tekrar sayma, fazla sayma ve az sayma gibi problemler ile karşılaşmıştır. Bunun sebebi iki sanal çizgi arasında nesnenin görüntü karesindeki ilerleme hızına bağlı olarak tekrar görülmesi ya da çok hızlı geçmesidir. Bundan dolayı geliştirilen yazılım fazla veya eksik saymaktadır. İki sanal çizgi arasından nesnelerin sayılması işleminde nesne üzerinde hakimiyet kurulamamaktadır. Tek çizgi ile yapılan çalışmada nesnelerin bir önceki ve bir sonraki görüntü kareleri arasında koordinat değerlerine göre uzaklık değerlerinin bulunması, uzaklık değerine göre çizgiden geçip geçmediğinin kontrol edilmesi %100 sayma sonucu vermiştir. Şekil 5a'da iki sanal çizgi arasından geçen sınırlayıcı dikdörtgen koordinat değerlerine göre son nesne sayma işlemi, Şekil 5b'de iki sanal çizgi arasından geçen ağırlık merkezi koordinat değerlerine göre son nesne sayma işlemi, Şekil 5c'de tek sanal çizgi üzerinden geçen sınırlayıcı dikdörtgen koordinat değerlerine göre son nesne sayma işlemi, Şekil 5d'de tek sanal çizgi üzerinden geçen ağırlık merkezi koordinat değerlerine göre son nesne sayma işlemi verilmiştir.



**Şekil 5.** (a) İki sanal çizgi arasından geçen sınırlayıcı dikdörtgen koordinat değerlerine göre son nesne sayma işlemi (b) İki sanal çizgi arasından geçen ağırlık merkezi koordinat değerlerine göre son nesne sayma işlemi (c) Tek sanal çizgi üzerinden geçen sınırlayıcı dikdörtgen koordinat değerlerine göre son nesne sayma işlemi (d) Tek sanal çizgi üzerinden geçen ağırlık merkezi koordinat değerlerine göre son nesne sayma işlemi

*Figure 5. (a) The process of counting the last object with respect to the bounding rectangular coordinate values passing between the two virtual lines (b) The process of counting the last object with respect to the centroid coordinate values passing between the two virtual lines (c) The process of counting the last object with respect to the bounding rectangular coordinate values passing over the single virtual line (d) The process of counting the last object with respect to the centroid coordinate values passing over a single virtual line*



Nesne sayma çözümleri hem zaman açısından hem de insan gücü bakımından maliyetleri azalttığı için tercih edilen endüstriyel uygulamalardır. Boyut olarak küçük, adet bakımından çok fazla olan ürünlerin sayılması ve paketlenmesi için kamera ve görüntü işleme tabanlı sistemler pratik kullanımlar sunmaktadır. Ancak ürün boyutunun çok küçük olması, malzemenin kamera önünden çok hızlı geçmesi, pozlama süresi, çözünürlük, fps oranı, malzemelerin ilerlerken üst üste binmesi veya yapışık ilerlemesi, nesne kadrage girip çıkıncaya kadar tek bir kimliğe sahip olması gerekliliği, segmentasyon, aydınlatma gibi birçok problemi beraberinde getirmektedir. Birçok farklı parametrenin görüntü işleme algoritma performansını doğrudan etkilediği bu durumlarda sadece ilgilenen nesneye özel bir uygulama tasarlanması veya nesneye göre adaptif bir sistem tasarlanması zorunluluğu ortaya çıkmaktadır. Bu çalışmada, hareketli nesnelerin video görüntüleri kaydedilerek sayma işlemi gerçekleştirilmiş, ileri uygulamalar için bir ön çalışma yapılmıştır. Önerilen iki farklı koordinat noktasına ve uzaklık ölçme kriterine göre tek bir sanal çizgi kullanılarak %100 sayma işlemi gerçekleştirilmiştir. Gelecek çalışmalarda serici hızı ve banttı ilerleyen nesne sayısı arttırılarak gerçek zamanlı çözümler gerçekleştirilecektir. Tensorflow ve YOLO gibi derin öğrenme kütüphanelerinde bulunan nesne sayma ve nesne tespiti API'leri kullanılarak uygulama geliştirilmesi ve bu uygulamaların hızlı sonuç vermesi için GPU teknolojisi kullanan NVIDIA Jetson kartlarında yazılımın koşturulması hedeflenmektedir.

#### KATKI BELİRTME (ACKNOWLEDGEMENT)

Bu çalışma Necmettin Erbakan Üniversitesi Bilimsel Araştırma Koordinatörlüğü 181719004 numaralı proje ile desteklenmiştir.

#### KAYNAKLAR (REFERENCES)

- Aich, S., Stavness, I., "Leaf counting with deep convolutional and deconvolutional networks", *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Venice, Italy, 2080-2089, 22-29 Oct. 2017.
- Aich, S., Josuttis, A., Ovsyannikov, I., Strueby, K., Ahmed, I., Duddu, H. S., Pozniak, C., Shirliff, S., Stavness, I., "Deepwheat: Estimating phenotypic traits from crop images with deep learning", *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, 323-332, 12-15 March 2018.
- Antonini, G., Thiran, J.-P. J. I. T. o. C., Technology, S. f. V., 2006, "Counting pedestrians in video sequences using trajectory clustering", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 8, pp. 1008-1020.
- Arteta, C., Lempitsky, V., Noble, J. A., Zisserman, A., "Learning to detect cells using non-overlapping extremal regions", *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Berlin, Heidelberg, 348-356, 2012.
- Barbedo, J. G. A., "Counting clustered soybean seeds", *2012 12th International Conference on Computational Science and Its Applications*, Salvador, Brazil, 142-145, 2012.
- Baygin, M., Karakose, M., Sarimaden, A., Akin, E. J. a. p. a., 2018, "An image processing based object counting approach for machine vision application", *arXiv preprint arXiv:1802.05911*.
- Chan, A. B., Liang, Z.-S. J., Vasconcelos, N., "Privacy preserving crowd monitoring: Counting people without people models or tracking", *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 1-7, 2008.
- Chan, A. B., Vasconcelos, N. J. I. T. o. I. P., 2011, "Counting people with low-level features and Bayesian regression", *IEEE Transactions on Image Processing*, Vol. 21, No. 4, pp. 2160-2177.
- Chen, K., Gong, S., Xiang, T., Change Loy, C., "Cumulative attribute space for age and crowd density estimation", *Proceedings of the IEEE conference on computer vision and pattern recognition*, Portland, OR, USA, 2467-2474, 2013.

- Lin, Z., Davis, L. S. J. I. T. o. P. A., Intelligence, M., 2010, "Shape-based human detection and segmentation via hierarchical part-template matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 4, pp. 604-618.
- Mazurek, P., YOLO-Object-Counting-API, <https://github.com/tugot17/YOLO-Object-Counting-API>, Ziyaret Tarihi: 28.09.2020
- Onoro-Rubio, D., López-Sastre, R. J., "Towards perspective-free object counting with deep learning", *European Conference on Computer Vision*, Amsterdam, Netherlands, 615-629, 11-14 October, 2016.
- OpenCV, 2020a, Image Thresholding, [https://docs.opencv.org/3.4/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html), Ziyaret Tarihi: 28.09.2020.
- OpenCV, 2020b, Motion Analysis, [https://docs.opencv.org/master/de/de1/group\\_\\_video\\_\\_motion.html](https://docs.opencv.org/master/de/de1/group__video__motion.html), Ziyaret Tarihi: 28.09.2020.
- OpenCV, 2020c, Contours : Getting Started, [https://docs.opencv.org/master/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html), Ziyaret Tarihi: 28.09.2020.
- OpenCV, 2020d, Structural Analysis and Shape Descriptors, [https://docs.opencv.org/3.4/d3/dc0/group\\_\\_imgproc\\_\\_shape.html#ga103fcbda2f540f3ef1c042d6a9b35ac7](https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#ga103fcbda2f540f3ef1c042d6a9b35ac7), Ziyaret Tarihi: 28.09.2020.
- Özlu, A., 2020, Tensorflow Object Counting API, [https://github.com/ahmetozlu/tensorflow\\_object\\_counting\\_api](https://github.com/ahmetozlu/tensorflow_object_counting_api), Ziyaret Tarihi: 28.09.2020.
- Öztürk, Ş., Özkaya, U., 2020, "Skin Lesion Segmentation with Improved Convolutional Neural Network", *Journal of Digital Imaging*.
- Pandit, A., Rangole, J., Shastri, R., Deosarkar, S., "Vision system for automatic counting of silkworm eggs", *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 1-5, Chennai, India, 2014.
- Perera, P., Fernando, W., Herath, H., Godaliyadda, G., Ekanayake, M., Wijayakulasooriya, J., "A generic object counting algorithm under partial occlusion conditions", *IEEE 8th International Conference on Industrial and Information Systems*, Peradeniya, Sri Lanka, 554-559, 17-20 Dec. 2013.
- Rabaud, V., Belongie, S., "Counting crowded moving objects", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA , 705-711, 17-22 June 2006.
- Raman, M. S., Sukanya, M., 2012, "A novel labelling algorithm for object counting", *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, Coimbatore, India , 1-7, 26-28 July 2012.
- Ryan, D., Denman, S., Fookes, C., Sridharan, S., "Crowd counting using multiple local features", *Digital Image Computing: Techniques and Applications*, Melbourne, VIC, Australia, 81-88, 1-3 Dec. 2009.
- Thammasorn, P., Boonchu, S., Kawewong, A., "Real-time method for counting unseen stacked objects in mobile", *IEEE International Conference on Image Processing*, Melbourne, VIC, Australia, 4103-4107, 15-18 Sept. 2013.
- Wang, M., Wang, X., "Automatic adaptation of a generic pedestrian detector to a specific traffic scene", *CVPR 2011*, Providence, RI, USA , 3401-3408, 20-25 June 2011.
- Zhang, C., Li, H., Wang, X., Yang, X., "Cross-scene crowd counting via deep convolutional neural networks", *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, MA, USA , 833-841, 7-12 June 2015.
- Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y., "Single-image crowd counting via multi-column convolutional neural network", *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 589-597, 27-30 June 2016.