

## **ÖZET**

Haar cascade yöntemini kullanarak bir fotoğraf içerisinde kaç adet insan bulunduğunu yüz tanıma sistemi ile bulan bir uygulamadır. Bu uygulama bulunan yüzleri tek tek kullanıcıya bir kare içerisinde göstermekte aynı zamanda bu kareleri ana fotoğraftan keserek kullanıcıya başka bir alanda vurgulanmıştır. Haar cascade yöntemi ara yüze yüklenen resmi ilk olarak gray scale, daha sonra bitmap'e çevirdikten hemen sonra pikseller arası çeşitli yapıları arayarak insan yüzünde bulunan göz, burun ve ağız yapılarını çıkarılmamaktadır. Bunun için daha önceden tanımlanmış data seti kullanarak karşılaştırmalar yaptıktan sonra sonuçları sergilemektedir. Aynı zamanda bu uygulama haar cascade algoritmasındaki değişkenleri ara yüz vasıtasıyla değiştirerek kolaylıkla bize uygun ayarların oluşmasını sağlamaktadır

**Anahtar Kelimeler:** Haar cascade, göz, burun, ağız, yüz, insan sayma

## **İÇERİK**

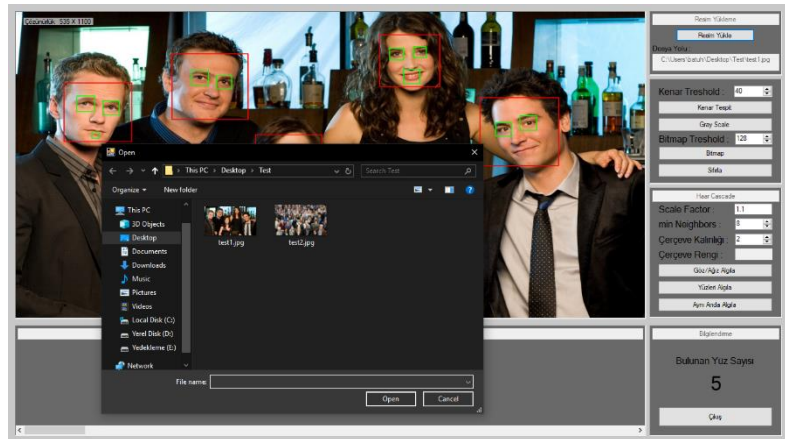
Yüz tanıma, 'bilgisayarla görme' teknolojisi altında sınıflandırılan bir uygulama türüdür. Gerçek zamanlı olabileceği gibi daha önce kaydedilmiş bir video veya çekilmiş bir fotoğraf üzerinden de yüz tanıma işlemi gerçekleştirilebilir.

Projemizde daha önceden çekilmiş bir fotoğrafı kullanarak, üzerinde GrayScale'ye çevirme ve kenar tespit etme işlemleri gerçekleştirilerek görüntüde kaç adet yüz bulunduğunu görebilmekteyiz.

Bunlar dışında görüntüyü GrayScale, Bitmap ve Kenarları tespit edilmiş olarak görebilmekteyiz. Bulunan yüzün etrafındaki çerçevenin boyutunu ve rengini değiştirebilmekteyiz. Bulunan her bir yüzü, programın sağ tarafında ana resimden kesilerek oluşturulan yeni resimler şeklinde sergilemekteyiz. Burada her bir yüzü kaçınıcı sırada bulunduğunu ve detaylarını gözlemleyebilmekteyiz. Aynı zamanda uygulamanın farklı bir alanında bu kişiler arasında bulmak istediğimiz kişinin var olup olmadığını görebildiğimiz bir alan bulunmakta. Bu alanın kullanımında ilk önce sisteme bir fotoğraf eklenmekte daha sonra bu fotoğraf üzerinde yüz tanıma işlemi yapılmakta ardından bulmak istediğimiz kişinin fotoğrafı seçilmekte ve son olarak sağ taraftaki fotoğraflardan bu aramayı gerçekleştirmektedir.

## **Görüntü Ekleme**

Sağ üst köşede bulunan Resim Yükle butonuna tıklayarak JPG, JPEG, PNG formatında görüntüler üzerinde işlemler yapmak üzere programa eklenir yani open file dialog penceresi açılarak yüklenilmesi istenilen resmi seçer.



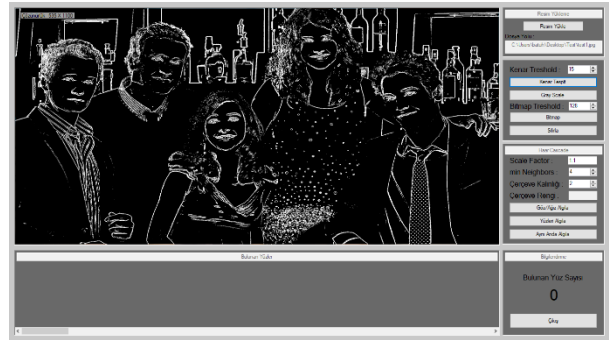
## GrayScale ve Bitmap

Eklenilen görüntüde her bir pikselin RGB değerlerinin ortalaması alınarak, gri ve tonları olan 8 bitlik GrayScale dönüştürülür. 1 bit olan ve sadece siyah ile beyaz renklerinin bulunduğu görüntü biçimine Bitmap denir. GrayScale üzerinden işlemler gerçekleştirilir. Gray image değerlerinin belirlenen bir eşik değerinin altında veya üstünde olmasına göre 0 ve 1 değerleri yani siyah beyaz değerleri verilmektedir.



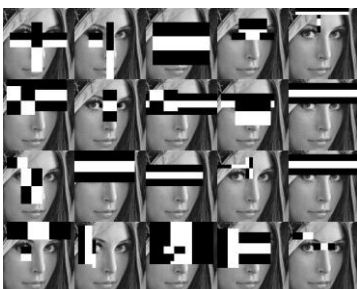
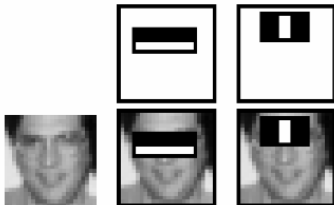
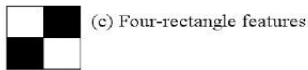
## Kenar Tespit

Görüntüde bütün pikseller tek tek hesaplanmak üzere en sol üstten başlanarak 3x3lük matrisler alınır. Ortadaki pikselin x ve y'si baz alınarak etrafındaki x'ler ve y'ler ile hesaba sokulurlar. Yapılan hesapta İpikselx ve İpiksely formülleri uygulanır. Ardında kenarları tespit etmek için de İpikselgradient uygulanarak bulunan değer ile threshold karşılaştırılması yapılır. Değer Threshold'dan büyükse 1 yani kenarlık var kabul edilir, eğer küçükse 0 yani kenarlık yok kabul edilir.

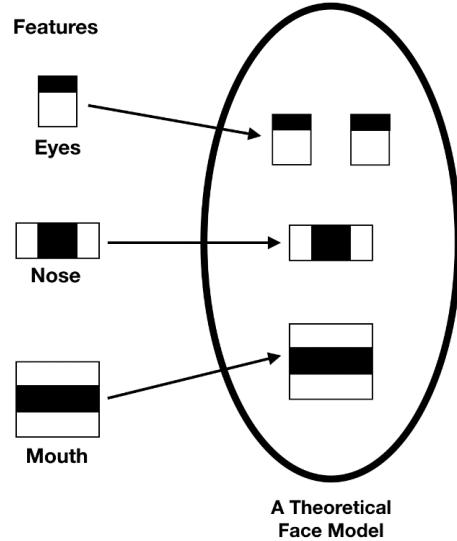


## Haar Cascade çalışma prensibine örnekler.

Kenarların belirlenmesi



Belirlenen kenarların yüz hatlarını bulmak için kullanılması.



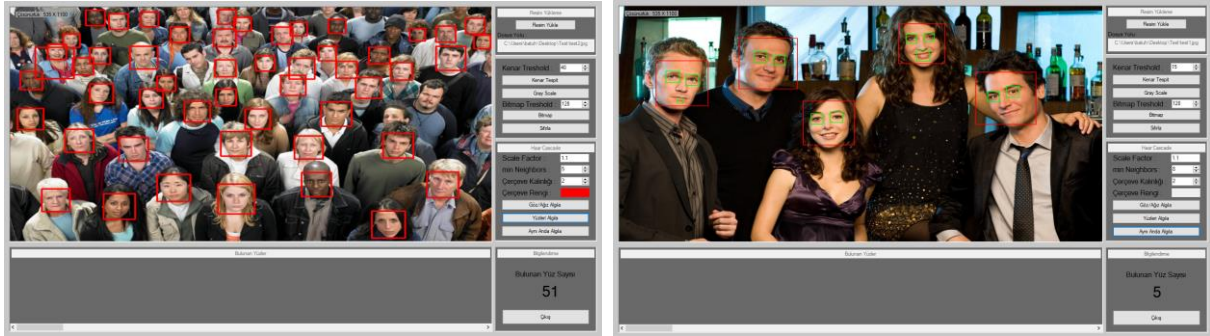
Yüz hatları için uygun olanların görüntüdeki yüz üzerinde denenmesi.

## Haar Cascade Face ve Eye Detection

EmguCV içerisinde bulunan haar cascade face detection sınıfı yardımıyla, fotoğraf içerisinde bulunan yüz hatlarını tespit edip, bunun yüz olup olmadığına bakan yüz tespiti algoritmasını kullanmaktayız. Aynı şekilde EmguCV içerisinde bulunan haar cascade eye detection sınıfı yardımıyla, fotoğraf içerisinde bulunan göz hatlarını tespit edip bunun göz olup olmadığına bakan yüz tespit algoritmasını kullanmaktayız.

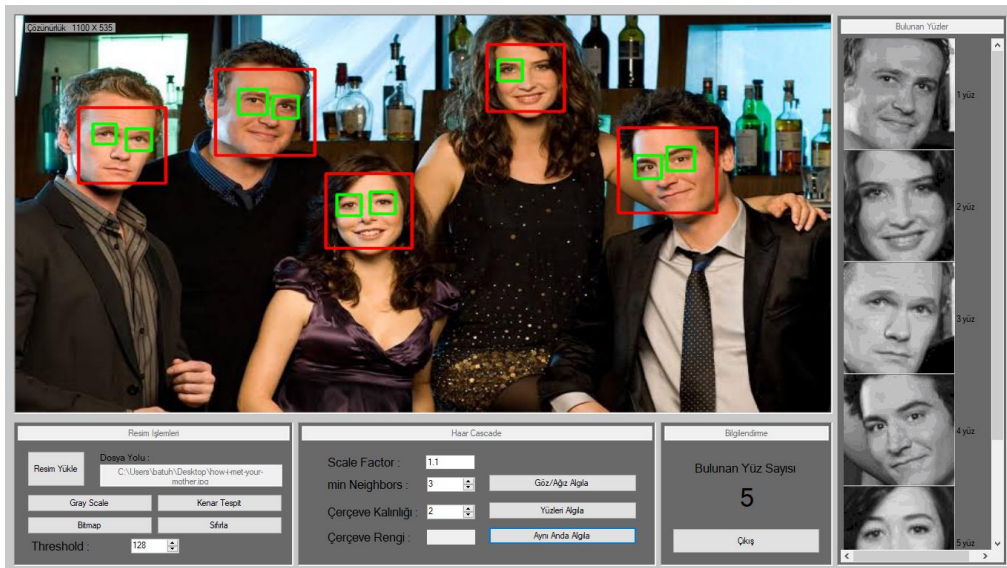
## Fotografda Kaç Adet Yüz Olduğunun Bulunması

Yüklenen fotoğrafta yüz algılama veya göz algılama butonlarına tıklandığında fotoğraf üzerine bulunan yüzleri belirtmek adına çeşitli kare şekiller ile gösterilir. Arka planda bu çizim yapılırken aynı anda bir sayma işlemi gerçekleştirilir. Bu sayma işlemi fotoğrafa kaç adet kare çizildiğini yani fotoğrafta kaç adet yüz olduğunu saymaktadır. Bu sayma işlemi bittikten sonra kullanıcıya arayüz yardımı ile sayma işleminin sonucu gösterilir.



## Kişilerin Ayır Ayır Sergilenmesi

Ara yüz üzerinden sisteme fotoğraf yüklenmesi sonrasında gerçekleştirebildiğimiz haar cascade face ve eye detection işlemi sırasında bu işlem arka planda gerçekleşmektedir. Bu işlem aslında bir fotoğraf üzerinden belirli alanların kesilmesine dayanmaktadır. Yüz ve gözleri bulduktan sonra bunları kullanıcıya göstermek amacı ile yüz ve göz çevresine belirli kareler çizilmekte. Bu işlemde esas alacağımız yüz hatlarının etrafına çizdiğimiz kareler. Bu karelerin koordinat düzlemi üzerindeki değerlerinin alınarak, işlem yapılan resmin bir kopyası oluşturularak bu koordinat değerleri çerçevesinde bir kare kesme işlemi uygulanmaktadır. Bu kesme işlemi her yüz için tekrar tekrar gerçekleştirilmektedir. Her kesim sonra ise sergileme alanına yerleştirilmektedir. Fotoğraf üzerinde birçok yüz bulunabileceğinden dolayı bu sergileme alanı bir list view olarak seçilerek sonsuz sayıda fotoğraf eklenebilmesi sağlandı. Alanın boyutlarından çıkması doğrultusunda scroll bar ile bu fotoğraflar arasında gezerek kişileri daha detaylı görebilmekteyiz.



**Unit Testler :** Uygulama bu aşamada haar cascade algoritmasının ne kadar doğruluk ile tahmin yaptığını anlamak ve değişken değerlerinin oturtulması adına birçok testten geçirildi. Aynı zamanda programda oluşabilecek hataların da bu testler sayesinde geliştirilmesi sağlandı. Bu testler farklı ışık düzeylerine sahip, farklı kişi sayılarına sahip, farklı çözünürlüklere sahip ve farklı ortamlara sahip birçok görüntü ile gerçekleştirildi.

#### **Projede Kullanılan Teknolojiler :**

- i. C# : Proje boyunca yapılacak işlemlerin yazılacağı kodlama dili..
- ii. EmguCV : Görüntü işleme işlemlerinin yapılması için gerekli bir modül.
- iii. Visual Studio : C# kodlarının derlenmesi için gerekli IDE.
- iv. Haar Cascade : Yüz tanıma işleminin yapıldığı algoritma.

#### **KAYNAKÇA**

1. Gökmen, M., Kurt, B., Kahraman, F., & Çapar, A. (2007). Çok Amaçlı Gürbüz Yüz Tanıma. İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü, Tübitak Projesi, Proje, (104E121).
2. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), 71-86.
3. Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, 43(1), 3-31.
4. Benvenuto, F., & Marani, A. (2000). Neural networks for environmental problems: data quality control and air pollution nowcasting. Global NEST: The International Journal, 2(3), 281-292.
5. Boznar, M., Lesjak, M., & Mlakar, P. (1993). A neural network-based method for short-term predictions of ambient SO2 concentrations in highly polluted industrial areas of complex terrain. Atmospheric Environment. Part B. Urban Atmosphere, 27(2), 221-230.
6. Cheng, B., & Titterton, D. M. (1994). Neural networks: A review from a statistical perspective. Statistical science, 2-30.
7. Comrie, A. C. (1997). Comparing neural networks and regression models for ozone forecasting. Journal of the Air & Waste Management Association, 47(6), 653-663.
8. Goldstein, A. J., Harmon, L. D., & Lesk, A. B. (1971). Identification of human faces. Proceedings of the IEEE, 59(5), 748-760
9. Haig, N. D. (1985). How faces differ—A new comparative technique. Perception, 14(5), 601-615
10. Jain, A. K., & Kumar, A. (2010). Biometrics of Next Generation: An Overview. proceedings Second Generation Biometrics' Springer.
11. Kaufman, G. J., & Breeding, K. J. (1976). The automatic recognition of human faces from profile silhouettes. IEEE Transactions on systems, Man, and Cybernetics, (2), 113-121.
12. Jain, A. K., & Li, S. Z. (2011). Handbook of face recognition. New York: Springer.
13. Martinez, A. M. (1998). The AR face database. CVC technical report
14. Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and modular eigenspaces for face recognition. In CVPR (Vol. 94, pp. 84-91).
15. Rhodes, G. (2013). Looking at faces: First-order and second-order features as determinants of facial appearance. Perception, 42(11), 1179-1199
16. Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. Josa a, 4(3), 519-524.