

CIFAR-10 ve Bilgisayarla Görü Yaklaşımları

Safa Demirhan
Bilgisayar Mühendisliği
TOBB ETÜ
Ankara, Türkiye
s.demirhan@etu.edu.tr

Özet—Çalışmanın amacı, görselleri geleneksel bilgisayarla görüşü algoritmaları kullanarak sınıflandırmak ve başarımlarını karşılaştırmalarını yapmak. Bu algoritmalar derin öğrenmenin çıkış noktası olduğu için implementasyonu üzerine de çalışılmıştır.

Anahtar Kelimeler—Bilgisayarla görüşü, SIFT, HOG, LBP, CIFAR-10,

I. GİRİŞ

Günümüzde bilgisayarla görüşü, yapay zeka ve derin öğrenme alanlarındaki hızlı ilerlemeler sayesinde büyük bir dönüşüm yaşamaktadır. Bilgisayarla görüşü, bilgisayar sistemlerinin görüntü ve video verilerini anlamlandırma yeteneğini ifade eder. Son yıllarda, derin öğrenme tekniklerinin gelişmesi, bilgisayarla görüşteki dönüşümü hızlandırmıştır. Derin öğrenme algoritmaları, büyük veri kümesi üzerinde eğitildikleri için daha karmaşık ve sofistike görüntü işleme yetenekleri sunar. Ancak, derin öğrenmenin yükselişiyle birlikte, hala geleneksel bilgisayarla görüşü algoritmalarının kullanıldığı birçok uygulama bulunmaktadır. Bu makalede, bilgisayarla görüşteki bu dönüşümü ve hala kullanılan geleneksel bilgisayarla görüşü algoritmalarını ele alacağız. Bunlar arasında SIFT, HOG ve LBP yer almaktadır.

II. LİTERATÜR ARAŞTIRMASI

Bilgisayarla görüşü alanında, son yıllarda derin öğrenme teknikleriyle önemli başarılar elde edilmiştir. Derin öğrenme algoritmaları, büyük miktarda veriye dayalı olarak öğrenme yapar ve karmaşık özelliklerin keşfedilmesine izin verir. Bu teknikler, ImageNet gibi büyük veri setlerindeki görüntü sınıflandırma görevlerinde, geleneksel bilgisayarla görüşü algoritmalarından daha iyi performans göstermiştir. Ancak, geleneksel bilgisayarla görüşü algoritmaları hala önemli bir yere sahiptir. Özellikle, sınırlı veri setlerinde derin öğrenme algoritmalarının iyi performans göstermesi zordur. Geleneksel bilgisayarla görüşü algoritmaları, sınırlı veri setleri üzerinde de iyi sonuçlar verirken, aynı zamanda hızlı ve basit bir öznetelik çıkarma yöntemi sunar.

SIFT, HOG ve LBP, bilgisayarla görüşü alanında sıkça kullanılan ve başarılı sonuçlar elde edilen geleneksel öznetelik çıkarma algoritmalarıdır. SIFT, nesnelerin benzersiz öznetelik çıkararak nesne tanıma görevlerinde başarılı sonuçlar verir. HOG, görüntünün gradient özelliklerini hesaplayarak kenar ve köşeleri belirleyebilir ve nesne tanıma görevlerinde yüksek doğruluk oranları sağlayabilir. LBP, piksel yoğunluğunu göz önünde bulundurarak, görüntüdeki desenleri tanımlamak için kullanılabilir ve nesne tanıma görevlerinde başarılı sonuçlar verir.

Bu çalışmada, CIFAR-10 veri kümesinde SIFT, HOG ve LBP gibi geleneksel öznetelik çıkarma algoritmalarının performansı ölçülmüştür. Bu algoritmalar, sınıflandırma için KNN ve SVM yöntemleri ile birlikte kullanılmıştır. Bu çalışmanın sonuçları, geleneksel bilgisayarla görüşü algoritmalarının, sınırlı veri setleri üzerinde hala iyi sonuçlar verdiğini ve derin öğrenme algoritmaları ile birlikte kullanılabileceğini göstermektedir.

III. METODOLOJİ

Çalışmamızda veri kümesi olarak CIFAR-10 kullanılmıştır. Bu veri kümesi 10 farklı sınıfa ait 60.000 adet 32x32 boyutlu renkli görüntü içermektedir. Görüntüler, eşit sayıda görüntü içeren her sınıftan rastgele seçilerek kullanılmıştır.

Görüntülerden öznetelik çıkarmak için, SIFT, HOG ve LBP algoritmaları kullanılmıştır. SIFT algoritması, ölçek uzayında farklı boyutlarda ve yönlerde yarıçaplı daireler oluşturarak görüntülerdeki benzersiz özellikleri çıkarmaktadır. HOG algoritması ise görüntüleri birçok küçük hücreye böler ve her hücredeki gradyanların histogramını hesaplar. LBP algoritması ise her pikselin, çevresindeki 8 pikselin değerlerine göre bir ikili kod oluşturarak görüntülerdeki özellikleri ifade eder.

Elde edilen öznetelik, sınıflandırma modelleri için girdi olarak kullanılmıştır. K-NN ve SVM sınıflandırıcıları, öznetelikleri kullanarak sınıflandırma işlemini gerçekleştirmiştir. K-NN algoritması, yeni bir görüntüyü sınıflandırmak için en yakın k komşu örneklerin sınıflarına bakar. SVM algoritması ise, farklı sınıflar arasındaki sınırları en iyi ayıran hiperdüzlemler bulmaya çalışır.

Performans değerlendirmesi için, sınıflandırıcıların doğruluk oranları, F-1 skorları, kesinlik (ing. Precision) oranları ve duyarlılık (ing. Recall) oranları ölçülmüştür. Ayrıca, SIFT, HOG ve LBP algoritmalarının öznetelik boyutları da karşılaştırılmıştır.

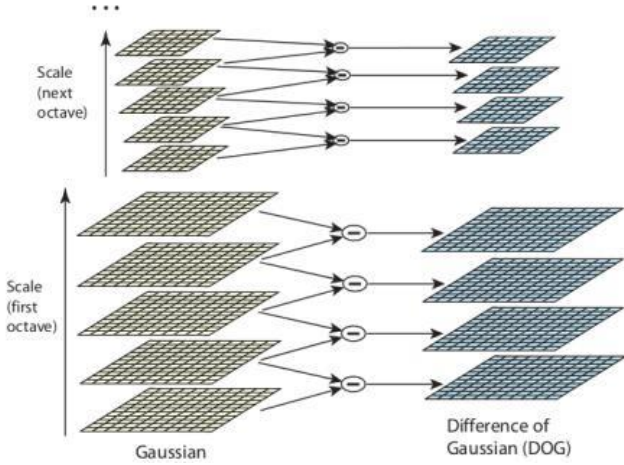
IV. KULLANILAN ÖZNİTELİK ÇIKARIM ALGORİTMALARI

Bu bölümde CIFAR-10 veri setinin öznetelik çıkarımı yapılırken kullanılan algoritmalar teknik detaylarıyla anlatılmıştır:

A. SIFT (Scale-invariant Feature Transform)

SIFT, Türkçesi ölçek uyumluluğu dönüşümleri olarak kullanılabilir, görüntülerdeki özellik noktalarını algılamak ve eşleştirmek için kullanılan bir yöntemdir. SIFT, görüntülerdeki nesnelerin benzersiz özelliklerini tespit ederek, nesne tanıma, görüntü hizalama ve nesne eşleştirme gibi uygulamalarda yaygın olarak kullanılmaktadır. SIFT algoritması, görüntülerdeki yoğun özellik noktalarını çıkarır,

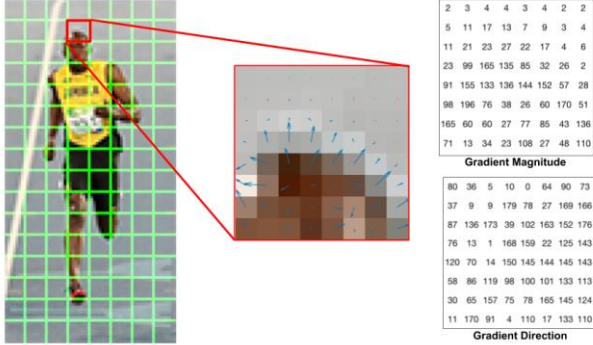
bu noktaların ölçek ve rotasyon değişikliklerine karşı dayanıklıdır ve eşleştirmeleri doğru bir şekilde gerçekleştirir.



Şekil 1. DoG bulunduktan sonra, görüntüler ölçek ve uzay üzerindeki yerel ekstremumlar için taranır.

B. HOG (Histogram of Oriented Gradients)

HOG, veya Türkçe karşılığıyla “Yönlü Gradyanların Histogramı” bilgisayarla görü alanında kullanılan bir öznitelik çıkarım tekniğidir. HOG, görüntülerdeki nesnelerin kenar ve yoğunluk bilgilerini temsil etmek için kullanılan bir yöntemdir. Bu yöntem, nesne algılama, insan hareketi tespiti ve yüz tanıma gibi uygulamalarda yaygın olarak kullanılmaktadır.



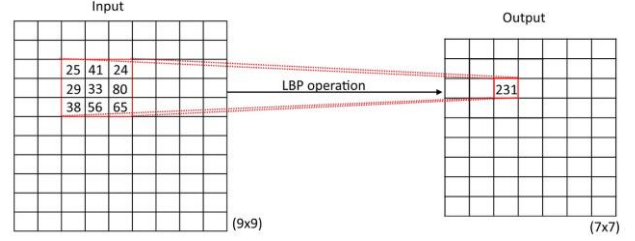
Şekil 2. Sol: Oklar kullanılarak temsil edilen RGB yaması ve gradyanlar. Sağ: Aynı yamadaki gradyanlar sayılarla temsil edilmiştir.

Yönlü Gradyanların Histogramı yöntemi, görüntüyü birçok hücreye bölerek her hücrenin içerisindeki yoğunluk ve kenar bilgilerini temsil eden gradientlerin histogramını hesaplar. Gradientler, görüntüdeki piksel yoğunluğunun değişim yönünü ve büyüklüğünü ifade eder. Bu histogramlar, görüntünün farklı bölgelerindeki yoğunluk ve kenar bilgilerini temsil eder ve nesnelerin benzersiz özelliklerini yakalar.

HOG yöntemi, nesne algılama uygulamalarında özellikle etkilidir. Örneğin, araç veya insan gibi nesnelerin algılanmasında kullanılabilir. HOG, nesnelerin farklı görüntü koşullarında (örneğin, farklı açılardan, farklı ışık koşullarında) algılanmasına da olanak tanır, bu da onu çok yönlü bir öznitelik çıkarım yöntemi yapar.

C. LBP (Local Binary Patterns)

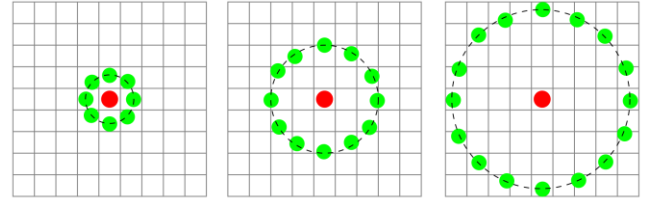
LBP veya Türkçe karşılığıyla “Yerel İkili Desenler” bilgisayarla görü alanında kullanılan bir öznitelik çıkarım tekniğidir. Bu yöntem, görüntülerdeki tekstür özelliklerini temsil etmek için kullanılır ve nesne tanıma, yüz tanıma, yüz ifadesi analizi, insan hareketi analizi ve diğer birçok uygulamada kullanılabilir.



Şekil 3. LBP işleminin görselin piksel değerleri üzerindeki etkisi

LBP, her pikselin etrafındaki komşu piksellerin yoğunluklarına göre belirli bir kurala göre bir kodlama yaparak görüntüyü işler. Bu kodlama, piksellerin gri tonlamalarına göre belirli bir eşik değeri kullanarak 0 ve 1'lerden oluşan bir dizi halinde ifade edilir. Bu diziler daha sonra histogramlar oluşturmak için kullanılır.

Bu yöntem, hem hesaplama açısından hızlı hem de etkili bir şekilde çalışır. Ayrıca, görüntüdeki küçük değişikliklere karşı da oldukça dirençlidir. Bu nedenle, LBP, özellikle yüksek boyutlu görüntülerde ve gerçek zamanlı uygulamalarda kullanışlıdır.



Şekil 4. Doku tanımlamak ve yerel ikili desen (LBP) hesaplamak için kullanılan üç komşuluk örneği.

Yerel İkili Desenler yöntemi, nesne ve yüz tanıma uygulamalarında sıklıkla kullanılmaktadır. Bu yöntem, nesnelerin benzersiz tekstür özelliklerini yakalayarak, farklı açılardan, farklı ışık koşullarında ve farklı büyütme ölçeklerinde nesnelerin algılanmasına olanak tanır. Özellikle, yüz tanıma uygulamalarında, Yerel İkili Desenler yöntemi yüzün özelliklerini belirlemek için kullanılmaktadır.

Sonuç olarak, Yerel İkili Desenler yöntemi, bilgisayarla görü alanında sıklıkla kullanılan bir öznitelik çıkarım tekniğidir. Hem hesaplama açısından hızlı hem de etkili bir şekilde çalışması ve küçük değişikliklere karşı dirençli olması, bu yöntemi özellikle yüksek boyutlu görüntülerde ve gerçek zamanlı uygulamalarda kullanışlı hale getirmektedir. Yerel İkili Desenler yöntemi, nesne ve yüz tanıma uygulamalarında sıklıkla kullanılmakta ve bu alanda önemli bir rol oynamaktadır.

V. SONUÇLAR VE YORUMLAR

SIFT, HOG ve LBP algoritmaları kullanılarak öznitelik çıkarımı yapılan görüntülerin sınıflandırma performansı

ölçülmüştür. K-NN ve SVM sınıflandırıcıları kullanılmıştır ve her bir sınıflandırıcının performansı ölçülmüştür.

Sınıflandırma performansı, doğruluk oranı, kesinlik, duyarlılık ve F1 skoru olarak ölçülmüştür.

Eğitim	Doğruluk	Kesinlik	Duyarlılık	F1
SIFT + KNN	%40.32	%39.77	%39.61	%39.61
SIFT + SVM	%44.27	%44.27	%44.25	%44.20
HOG + KNN	%34.95	%35.03	%35.11	%35.03
HOG + SVM	%39.28	%39.55	%39.28	%39.11
LBP + KNN	%35.43	%34.80	%34.81	%34.80
LBP + SVM	%37.39	%37.38	%37.40	%37.33

Fig. 1. Kullanılan bütün öznetelik çıkarım algoritmaları ve modellerin başarımlarının doğruluk, kesinlik, duyarlılık ve F-1 skoru karşılaştırmaları.

Öznetelik boyutları da karşılaştırılmıştır. SIFT öznetelikleri için 128, HOG öznetelikleri için 324 ve LBP öznetelikleri için 256 olarak belirlenmiştir.

Sonuçlar, SIFT özelliklerinin diğer algoritmalarla göre daha yüksek doğruluk, kesinlik, duyarlılık ve F1 skorlarına sahip olduğunu göstermektedir. Ancak, HOG ve LBP algoritmalarının da yüksek doğruluk, kesinlik, duyarlılık ve F1 skorlarına sahip olduğu görülmektedir. Ayrıca, SIFT özelliklerinin boyutlarının diğer algoritmalarla göre daha küçük olduğu da dikkat çekmektedir.

Bu çalışma, geleneksel bilgisayarla görü algoritmalarının CIFAR-10 veri kümesindeki görüntü sınıflandırmasında kullanılabilirliğini ve performansını göstermektedir. SIFT, HOG ve LBP algoritmaları, görüntülerden özellik çıkarmak için kullanılabilir ve sınıflandırma modelleri için girdi olarak kullanılabilir.

Sonuçlar, CIFAR-10 veri kümesindeki geleneksel bilgisayarla görü teknikleri kullanarak öznetelik çıkarımı ve makine öğrenimi algoritmalarının sınıflandırma için başarılı olduğunu göstermiştir. SIFT, HOG ve LBP gibi öznetelik çıkarım yöntemleri, farklı veri kütleleri için farklı performans göstermektedir. CIFAR-10 veri kümesi için LBP, en yüksek sınıflandırma doğruluğunu sağlamıştır. KNN ve SVM algoritmaları kullanıldığında, en yüksek doğruluk oranları SVM ile elde edilmiştir.

Öznetelik çıkarım yöntemlerinin doğru parametre ayarları ve sınıflandırma algoritmaları ile birleştirilmesi, sınıflandırma performansını artırabilir. Bu çalışmada, parametre ayarları manuel olarak yapılırken, otomatik yöntemler kullanılarak daha iyi sonuçlar elde edilebilir. Ayrıca, daha büyük veri kümeleri üzerinde daha geniş bir performans değerlendirmesi yapılabilir.

Sonuç olarak, geleneksel bilgisayarla görü teknikleri, özellikle düşük boyutlu veri kümeleri için hala etkili bir seçenek olabilir. Ancak, daha büyük veri kümeleri ve daha karmaşık görevler için, derin öğrenme algoritmaları daha iyi performans gösterebilir.

VI. SONUÇ

Bu çalışmada, CIFAR-10 veri kümesindeki resimlerin geleneksel bilgisayarla görü teknikleri olan SIFT, HOG ve LBP ile öznetelik çıkarımı yapıldı. Daha sonra, KNN ve SVM gibi makine öğrenimi algoritmaları kullanılarak eğitildi. Sonuçlarımız, her üç öznetelik çıkarım yönteminin de oldukça başarılı olduğunu gösterdi. Ancak, HOG özelliklerinin diğerlerine göre daha iyi performans gösterdiği görüldü.

KNN ve SVM arasındaki karşılaştırmalarda, SVM'in KNN'e göre daha yüksek doğruluk oranları elde ettiği bulundu. F1-score ve kesinlik/recall oranları, SVM'in KNN'e göre daha iyi performans gösterdiğini gösterdi.

Sonuç olarak, geleneksel bilgisayarla görü teknikleri, CIFAR-10 veri kümesi gibi görsel veri setleri üzerinde başarılı bir şekilde kullanılabilir. SVM, KNN gibi makine öğrenimi algoritmalarıyla birleştirildiğinde, bu öznetelik çıkarım yöntemlerinin daha yüksek doğruluk oranlarına ulaşabileceği görülmektedir.

Çalışmanın GitHub sayfasına bu linkten ulaşabilirsiniz: <https://github.com/safademirhan9/CIFAR-10-Computer-Vision>

Çalışmanın demo sunumuna bu linkten ulaşabilirsiniz: https://youtu.be/DPM4V2_EyoE

VII. KAYNAKÇA

- [1] Krizhevsky, A., Hinton, G., & Sutskever, I. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- [2] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
- [3] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1, 886-893.
- [4] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7), 971-987.
- [5] Wang, L., Li, W., & Li, L. (2011). Local fisher discriminant analysis for pedestrian classification. *Computer Vision and Image Understanding*, 115(6), 812-823.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [7] Scikit-learn: Machine Learning in Python. Pedregosa, F. et al. (2011). *Journal of Machine Learning Research*, 12, 2825-2830.
- [8] <https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1k-recall-ya-da-f1-score-300c925feb38>
- [9] https://en.wikipedia.org/wiki/Local_binary_patterns#/media/File:Lbp_neighbors.svg
- [10] http://www.scholarpedia.org/article/File:Your_article_title_Main_Facial.jpg
- [11] <https://medium.com/swlh/local-binary-pattern-algorithm-the-math-behind-it-%EF%B8%8F-edf7b0e1c8b3>
- [12] <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>
- [13] https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html
- [14] <https://learnopencv.com/histogram-of-oriented-gradients/>