

**TOBB ETÜ Yapay Zeka Mühendisliği Bölümü**  
**YAP 470 / BİL 570 Dönem Projesi 1. Ara Rapor**  
**2024-25 Yaz Dönemi**

Öğrenci Numarası: 221101006

Öğrenci Adı Soyadı: Beren Ünveren

<b>1. Proje ismi</b>
Makine Öğrenmesi ve Derin Öğrenme Yöntemleri ile Görüntü Tabanlı Yangın Tespiti
<b>2. Proje tanımı (kısaca özet) (projenin hangi problemi çözdüğü ve amacı)</b>
Yangınların erken tespiti; can ve mal kayıplarının önlenmesi, çevresel zararların azaltılması ve müdahale süreçlerinin etkinliği açısından kritik öneme sahiptir. Görüntü tabanlı yangın tespit sistemleri, kameraların yaygınlaşmasıyla birlikte geleneksel sensör tabanlı sistemlere güçlü bir alternatif sunmaktadır. Bu proje kapsamında ele alınan problem, bir dijital görüntü girdisi olarak bu görüntünün "yangın var" (fire) veya "yangın yok" (non-fire) şeklinde iki sınıftan birine atanmasıdır. Problem bir ikili görüntü sınıflandırma problemi olarak tanımlanmaktadır.
<b>3. Proje gereksinimleri ve projeden beklenen sonuç (projenin kısıtları, varsayımları, çalışma ortamı, girdileri, çıktıları, test etme yöntemleri ve metrikleri)</b>
Temel amaç görüntü tabanlı yangın tespiti problemi için iki farklı yöntem (ML-DL) geliştirmek, performanslarını karşılaştırmak, ve genel birkaç yöntemi de deneyip hepsinin performansını analiz etmektir. ML kısmında farklı özniteliklerin ve birbiriyle kombinasyonlarının performansları araştırılacak ve temel scikit modelleri eğitilecek ve optimizasyon yöntemleri keşfedilecektir. DL kısmında sıfırdan CNN tasarlanacak, eğitilecek ve hiperparametre optimizasyonları keşfedilecektir. Eğer non-fire görüntüsünün azlığının eğitimi olumsuz etkilendiği fark edilirse veri arttırımı veya dengelemesi denenecektir. Çalışma ortamı kişisel bilgisayarımdaki GPU ve Python'da numpy, pandas, OpenCV, Scikit-learn ve PyTorch (Araştırmalarda TensorFlow'dan daha esnek ve low level olduğu görülmüştür) kullanılacaktır. Girdiler daha öncesinde onayı alınmış iki datasetteki resimlerden çıkarılan öznitelikler ve çıktılar da yangının olup olmadığı şeklinde bir görüş olacak. Test etme yöntemleri; standart metrikler olan doğruluk,

kesinlik, duyarlılık ve f-1 skorunu karşılaştırmak olacaktır, karışıklık matrisleri incelenecektir.

**Kısıtlar:** Kullanılacak donanım kişisel bilgisayar GPU'sunda sınırlı kalacaktır. Kullanılabilecek kütüphaneler ve bu kütüphanedeki izin verilen fonksiyonlar sınırlıdır. Hazır model veya hazır feature extraction kullanmama zorunluluğu vardır. Görüntü işleme için belirtilen öznetelik mühendisliği yöntemlerinin ve modellerin projeye dahil edilme zorunluluğu vardır (MLP, boosting algoritması, SVM(RF yerine). Sıfırdan tasarlanacak en az bir CNN. Öznetelik mühendisliğinde PCA ve feature selection.) Projenin tahmini teslimine yaklaşık 35 gün olması istenen derinliğe ulaşamamasına sebep olabilir.

**Varsayımlar:** Sunulan datasetler yangın tespiti için yeterli çeşitlilikte ve kalitededir, renk ve doku analizine uygundur. Hareket tabanlı analiz yapılmayacağı için veriler art arda incelenmeyecek ve sıralı ilişki kurulmayacaktır. HOG ve LBP gibi öznetelik çıkarma yöntemleri için built-in algoritmalar incelenecek ve baştan yazılacaktır. Duman görüntüleri farklı renk ve doku özelliklerine sahiptir ve modelleri hem dumana hem yangına eğitmek çok meşakkatli olacaktır, bu sebeple D-Fire datasetindeki sadece duman olan görüntüler değerlendirmeye alınmayacaktır.

#### 4. Projede kullanılacak veri setleri (verilerin toplanma, işleme, etiketleme yöntemleri vb. nasıl olacak?)

Fire Dataset (Kaggle) doğrudan ikili sınıflandırmaya uygundur, fire\_images klasöründekiler 1 ile ve non\_fire\_images'dakiler 0 ile etiketlenecektir. Sınıf dengesizliği ihtimali bu datasette mevcuttur(755 yangın, 244 yangın olmayan) ve bu konu ayrıca incelenecektir (eğer sınıf dengesizliği optimize etmeye müsait başarımlar verirse temel görüntü arttırma teknikleri planlanmaktadır ve sonuçlar analiz edilecektir). D-Fire Dataset nesne tespitine uygun hazırlanmıştır ancak ikili sınıflandırmaya uyarlanabilir. İçinde "fire" veya "fire and smoke" bounding box'ı olan bir görüntü yangın etiketi(1) ile, "none" içerisindeki görüntü 0 ile etiketlenecektir.

#### 5. Literatür taraması (daha önce benzer projelerde kullanılan metotlar hakkında bilgiler) (kullanılan başlıca yöntemlerin açıklanması, karşılaştırılması ve değerlendirilmesi)

Yaklaşık 8 survey paper'ından toplanan yöntemler aşağıda verilmiştir. Survey'ler kaynakçada verilmiştir. Yöntemler içindeki avantaj/dezavantaj değerlendirmeleri projeye uygulanabilecek olanlara yapılmıştır.

**Görüntü İşleme + ML Yöntemleri:**

- ★ Renk tabanlı yöntemler: YcbCr ve HSV görülmüştür. Alev genellikle kırmızı-sarı tonlarında gezdiği için renk değerlerinden analiz yapmaya çalışılır. YCbCr de parlaklık değişiminden az etkilendiği için farklı ortam görüntülerinde daha iyi sonuç verir. En temel uzay olan RGB'nin kullanılmama sebebi parlaklığın ve renk bilgisinin birbirine bağlı olması ve farklı ortam ve şartlarda yangın tespitinde iyi sonuç vermemesidir.
  - Renk tabanlı yöntemlerin hesaplanmasının çok basit olmasının yanı sıra alev benzer renkler taşıyan ama alev olmayan görüntülerde sadece renge bağlı kalınması yetmemektedir ve false positive oranını arttırmaktadır.
- ★ Doku tabanlı yöntemler: LBP(local binary pattern) ve HOG(histogram of oriented gradients) görülmüştür. LBP piksel komşuluklarını kullanarak yerel dokuyu ifade eder, HOG ise görüntüdeki gradyan yönlerinin histogramını çıkararak şekil ve doku hakkında bilgi verir. LBP ve HOG için OpenCV'de hazır metotlar olmasına rağmen yasak olduğu için LBP ve HOG implementasyonları incelenebilir.
  - LBP alevin düzensiz yapısını algılamakta iyi olsa da lokal analiz olduğu için gürültüye hassas olma riski vardır. HOG LBP'ye kıyasla hesaplama maliyeti daha yüksek olan bir yöntemdir ama nesne şekil yakalamasında LBP'den daha başarılıdır.
- ★ Hareket tabanlı yöntemler: Optik akış ve arka plan çıkarma görülmüştür. Sıralı resimlerle çalışılmayacağı için taramaya daha fazla dahil edilmemiştir.
- ★ Klasik ML yöntemleri: SVM ve Random Forest görülmüştür. Renk, doku, hareket özellikleri özellik vektörü haline getirilip SVM ya da Random Forest gibi algoritmalara bu vektörler verilerek eğitim yapılmıştır.
  - SVM'in eğitim süresinin Random Forest'tan daha uzun olma riski vardır ancak SVM yüksek boyutlu uzaya çıkıldığında daha etkilidir.

#### DL Yöntemleri:

- ★ Görüntü sınıflandırma: CNN görülmüştür. Literatür araştırmasında kendi CNN'lerini yapanlar veya hazır mimariler kullananlar(AlexNet, VGGNet, ResNet, MobileNet vs. (transfer learning)) görülmüştür.
  - Hazır mimari projenin genel işleyişinde hız kazandırsa da probleme en uygun mimariyi aramak zordur ve sıfırdan mimari oluşturmak eğitim başarımı bakımından daha iyi olabilir. Sıfırdan mimari tasarımında eğitim daha uzun sürer ve overfitting riski yüksektir.
- ★ Nesne tespiti: YOLO, Faster R-CNN ve SSD görülmüştür (hazır mimariler, üçü de pretrained). Yangın var/yok probleminden de öteye geçip yangının olduğu yeri bounding box ile işaretlerler. YOLO hızıyla, Faster R-CNN detaycılığıyla göze çarpmıştır, SSD bu ikisinin ortasındadır.

- ★ Görüntü Bölütleme: U-Net ve DeepLab(hazır mimariler) görülmüştür. En hassas yöntem olarak bilinir ve görüntüdeki her pikseli yangın, duman ya da arka plan olarak sınıflandırır ve yangınlı bölgenin maskesini çıkarırlar.
- ★ Hibritler: CNN ve LSTM/GRU, 3DCNN görülmüştür(CNN ve LSTM/GRU konsept, 3DCNN hazır mimarı). CNN ve LSTM/GRU görüntüden mekansal, videodan zamansal bilgi alır, 3DCNN ise videoyu 3D veri olarak hem zamansal hem mekansal olarak işler. Sıralı resimlerle çalışılmayacağı için taramaya daha fazla dahil edilmemiştir.
- ★ Dikkat mekanizmaları: CBAM ve SENet görülmüştür(hazır mimari). Anladığım kadarıyla CNN'in temel yapısına bir katmanda ağırlıklar vererek istenen yapılara odaklatılarak background'un etkisini azaltıyor; self attention, channel attention ve spatial attention adında türleri var. Çok detaya girdiğimi düşünerek taramaya daha fazla dahil etmedim ama survey'lerde vardı.

**Genel karşılaştırmalar ve öznel seçimler:** Öznitelik çıkarımında renk ve dokunun beraber kullanmak istenmesinin sebebi öznitelik vektörünü daha da geliştirmek ve daha mantıklı sonuçlar almaya çalışılmasındandır. Sadece renge veya sadece dokuya odaklanmak etkili olmayacağından bu bilgiler beraber kullanılacaktır. ML'de MLP ve boosting algoritması zorunlu, SVM'in yüksek boyutlu vektörlerde bile verimli sonuç vermesi ve survey'de ML içinde en performansı yüksek model olarak görüldüğünden SVM seçilmiştir. Boosting algoritmalarından LightGBM seçilmesinin sebebi de gradient boost'un daha hızlı versiyonu olmasıdır. DL modellerinden CNN implementasyonu seçilmiştir. CNN'den sonra yeterli vakit kalırsa hibrit bir model (CNN'den öznitelik çıkarımı + ML) denenip son rapora onun da performans analizi eklenecektir.

#### **6. Yöntem 1 (yöntemin açıklaması, adımları, gerekli bölümlerinin uygulanma detayları)**

- ★ Veri ön işleme. Görüntüler okunur ve yeniden boyutlandırılır - aynı boyuta getirilirler (boyut performansı değiştirir ve hiperparametredir), piksel değerleri normalize edilir (0-1 veya standardizasyon - hiperparametredir) Renk uzayı dönüşümleri yapılır (RGB'den HSV'ye, RGB'den YCbCr'ye)
- ★ Öznitelik çıkarımı. Renk uzaylarından histogram hesaplanır, hangi kanalların daha ayırt edici olduğu gözlenir (H, S, Cb, Cr planlandı) Histogram boyutu ayarlanabilir (hiperparametredir). Görüntüler gri tonlamalı hale çevrilir ve LBP hesaplanır (LBP'deki varyant - basic, uniform, rotation invariant seçimi bir HP). HOG hesaplanır (window size, block size, cell size, bin sayısı HOG'un parametreleridir - HP). LBP ve HOG çıkarılırken gerekli araştırmalar yapılarak hazır fonksiyonlardan kaçınılıp baştan algoritma hazırlanacaktır.
- ★ Öznitelik mühendisliği. H/S histogramı, Cb/Cr histogramı, HOG vektörü, LBP histogramı tek bir vektörde birleştirilir ve bu vektöre boyut seçimi

(PCA) ve azaltımı (feature selection - korelasyon analizi ve RFE planlanmaktadır) ayrı ayrı uygulanacaktır. PCA'daki kaç ana bileşenin korunacağı ve feature selection'da korelasyon analizi ve RFE(ikisinde de kaç ana bileşen kalacağı (HP), RFE'de step ve estimator ayarlandığı araştırılmıştır, HP olabilir) ve öznitelik sayısı (hiperparametrelerdir). Hangi yöntemin en iyi sonuç verdiğini anlayabilmek için cross validation kullanılacaktır. (k-fold sayısı bir hiperparametredir)

- ★ Model eğitimi. Hazırlanan öznitelik vektörleri belirlenen ML modellerinde kullanılacaktır (MLP, LightGBM, SVM)
- ★ MLP tasarımı oluşturulacaktır. Batch Normalization ve Dropout'un opsiyonel olduğu ama BN'in stabilite ve öğrenme oranı yükselmesi ve Dropout'un aşırı öğrenmeyi azalttığı bilgisine ulaşılmıştır. Zaman kalırsa bu katmanların etkilerinin (olma/olmama durumundaki sonuçların) analiz edilmesi planlanmaktadır. MLP mimarisi tasarım taslağı:
  - Giriş Katmanı:
    - Nöron sayısı(öznitelik v. boyutu) ve fonksiyon(identity).
  - Gizli Katmanlar: (kaç tane olacağı hiperparametre): Dense -> Batch Normalization -> Aktivasyon Fonksiyonu -> Dropout
    - Nöron sayısı (her katman için ayrı olacak, HP)
    - Aktivasyon fonksiyonu: HP (ReLU ve Leaky ReLU - Araştırırken dying ReLU problem ile karşılaştım ve Leaky ReLU da deneyip sonuçları karşılaştırmak isterim ve raporda buna da yer vereceğimi düşünüyorum.)
  - Çıkış Katmanı:
    - Nöron Sayısı: 1 (ikili sınıflandırma olduğundan)
    - Aktivasyon Fonksiyonu: sigmoid
- ★ Hiperparametre optimizasyonu. Seçilen hp'ler görüntü boyutu, normalizasyon yöntemi, histogram boyutu, LBP varyantı, HOG parametreleri, MLP'de hidden layer size, activation, learning rate, max iteration; LightGBM'de objective, max depth, learning rate, subsamples, colsample by tree; SVM'de C vektörü, kernel, gamma vektörü, PCA'deki ana bileşen sayısı ile diğer feature elimination yöntemleri seçimidir (korelasyon analizi ve RFE - Korelasyon analizi için veri setinin korelasyon matrisi hesaplanacak ve belirlenen eşik değerine göre öznitelik seçimi yapılacaktır, RFE'de eğer izin verilirse scikit kullanılacaktır). Optimizasyonda GridSearchCV ve RandomizedSearchCV kullanılacaktır (RandomizedSearch en başta kullanılıp GridSearch ile fine-tuning yapımı planlanmaktadır) ve k-fold cross validation ile en iyi hiperparametre kombinasyonu elde edilecektir. Her model-öznitelik seti kombinasyonu ile performans karşılaştırılacaktır ve raporlanacaktır.

- ★ Modelleri değerlendirme ve raporlama. Değerlendirmede standart metrikler olan doğruluk, kesinlik, duyarlılık ve f-1 skoru ve confusion matrixleri verilecektir.

## 7. Yöntem 2 (yöntemin açıklaması, adımları, gerekli bölümlerinin uygulanma detayları)

- ★ Veri ön işleme. Görüntüler okunur ve yeniden boyutlandırılır - aynı boyuta getirilirler (boyut HP), piksel değerleri normalize edilir (0-1 veya standardizasyon - normalizasyon yöntemi HP)
- ★ CNN model mimarisi tasarımı. Katman türleri ve sıralaması belirlenecektir. Batch Normalization ve Dropout'un opsiyonel olduğu ama BN'in stabilite ve öğrenme oranı yükselmesi ve Dropout'un aşırı öğrenmeyi azalttığı bilgisine ulaşılmıştır. Zaman kalırsa bu katmanların etkilerinin (olma/olmama durumundaki sonuçların) analiz edilmesi planlanmaktadır. CNN mimarisi tasarım taslağı:
  - Katmanlar (Conv2D veya Dense) -> Batch Normalization -> Aktivasyon Fonksiyonu -> Dropout düzenini takip edecektir.
  - Giriş Katmanı:
    - Boyut: Görüntü boyutu (Genişlik, Yükseklik, Kanal Sayısı - HP)
    - Fonksiyon: Identity
  - Tekrarlayan Evrişim ve Havuzlama Blokları (N defa tekrar eder - N bir HP):
    - Evrişim Katmanı (Conv2D):
      - Filtre Sayısı: HP ve bloklar ilerledikçe arttırılacak
      - Filtre Boyutu: HP
      - Stride: HP
      - Padding: HP ( 'same', 'valid')
    - Batch Normalization Katmanı: (epsilon HP)
    - Aktivasyon Fonksiyonu: HP (ReLU ve Leaky ReLU - Araştırırken dying ReLU problemi ile karşılaştım ve Leaky ReLU da deneyip sonuçları karşılaştırmak isterim ve raporda buna da yer vereceğimi düşünüyorum.)
    - Dropout Katmanı: HP
    - Havuzlama Katmanı (MaxPooling2D):
      - Pool Size: HP (Genellikle (2, 2))
      - Stride: HP (Genellikle Pool Size ile aynı)
  - Flatten Katmanı: tekrarlayan bloklar bittikten sonra
  - Tam Bağlantılı Katmanlar (Dense - M defa tekrar eder - M bir HP):
    - Dense Katmanı: Nöron Sayısı: HP
    - Batch Normalization Katmanı (epsilon HP)
    - Aktivasyon Fonksiyonu: HP (ReLU ve Leaky ReLU)
    - Dropout Katmanı: HP
  - Çıkış Katmanı:

<ul style="list-style-type: none"> <li>■ Dense Katmanı: Nöron Sayısı: 1 (ikili sınıflandırma için)</li> <li>■ Aktivasyon Fonksiyonu: sigmoid</li> <li>★ Model Derleme (Training Configuration): Loss Function: Binary cross-entropy, Optimizer: Adam ve SGD (HP), optimizer'a ait hiperparametreler (learning rate, SGD için momentum) planlanmıştır.</li> <li>★ Model derleme ve eğitim. Eğitimde eğitim ve validasyon veri setleri ayrıştırılıp; batch size, epoch sayısı, öğrenme oranı belirlenecektir.</li> <li>★ Hiperparametre optimizasyonu (yukarıda belirtilen hp'lerin çoğu ya da zaman yeterse hepsi). Seçilen hp'lerin değerlerinin validasyon seti üzerinde doğrulanması. Her model-öznitelik seti kombinasyonu ile performans karşılaştırılacaktır ve belgelenecektir.</li> <li>★ Model değerlendirme ve raporlama, ardından zaman kalırsa hibrit yaklaşım.</li> </ul>
<p><b>8. Projenin ayrıntılı planı, akış şeması, veri işleyişi vs. (veri seti oluşturma, yöntem uygulama, karşılaştırma, geliştirme takvimi)</b></p>
<p>20 Haziran itibariyle 35 gün kaldığından dolayı yaklaşık 30-40 günlük bir plan yapılması daha uygun görülmüştür. İlk yöntemin ara raporuna bu rapordan sonra 14 gün vardır.</p> <ul style="list-style-type: none"> <li>★ Veri etiketleme, ön işleme fonksiyonlarının yazılması (2 gün)</li> <li>★ LBP/HOG implementasyonları &amp; debug (4 gün)</li> <li>★ Diğer feature çıkarma kodları (2 gün)</li> <li>★ Öznitelik mühendisliği/PCA/FS kodları &amp; ML model kodları (3 gün)</li> <li>★ ML hiperparametre optimizasyonu (2 gün)</li> <li>★ 2. rapor yazımı (1 gün)</li> <li>★ Temel CNN mimarisinin kodlanması &amp; debug (3 gün)</li> <li>★ CNN eğitim döngüsü (5 gün)</li> <li>★ CNN hiperparametre optimizasyonu (4 gün)</li> <li>★ Değerlendirme kodları &amp; karşılaştırma (3 gün)</li> <li>★ 3. rapor yazımı ve sunum / demo hazırlığı (3 gün)</li> <li>★ Makale formatında nihai rapor (kalan zaman - 12 gün)</li> </ul>
<p><b>9. Kaynakça</b></p>
<ul style="list-style-type: none"> <li>★ G. Cheng et al., “Visual fire detection using deep learning: A survey,” <i>*Neurocomputing*</i>, vol. 596, Art. no. 127975, 2024. doi: 10.1016/j.neucom.2024.127975.</li> </ul>

- ★ F. Bu and M. S. Gharajeh, "Intelligent and vision-based fire detection systems: A survey," *\*Image and Vision Computing\**, vol. 89, pp. 45-62, Sep. 2019, doi: 10.1016/j.imavis.2019.08.004.
- ★ S. B. Kukuk and Z. H. Kilimci, "Comprehensive Analysis of Forest Fire Detection using Deep Learning Models and Conventional Machine Learning Algorithms," *\*International Journal of Computational and Experimental Science and ENgineering\**, vol. 7, no. 2, pp. 84-94, 2021, doi: 10.22399/ijcesen.877793.
- ★ N. D. Ismail, R. Ramli, and M. N. Ab Rahman, "A Systematic Literature Review of Vision-Based Fire Detection, Prediction and Forecasting," *\*Jurnal Kejuruteraan\**, vol. 37, no. 1, pp. 191-218, Jan. 2025, doi: 10.17576/jkukm-2025-37(1)-13.
- ★ O. Mahmoud, A. Saad, and N. Nazih, "A Survey of Deep Learning Methods for Vision-Based Fire Detection and Localization," in *\*2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)\**, Cairo, Egypt, 2023, pp. 1-7, doi: 10.1109/MIUCC58832.2023.10278332.
- ★ G. Chen, T. Chandarasupsang, X. Luo, A. Tananchana, and L. Mu, "A Survey of Vision-based Fire Detection using Convolutional Neural Networks," in *\*2024 4th International Conference on Big Data Engineering and Education (BDEE)\**, Chengdu, China, 2024, pp. 29-36, doi: 10.1109/BDEE62400.2024.10515124.
- ★ R. N. Vasconcelos, W. J. S. Franca Rocha, D. P. Costa, S. G. Duverger, M. M. M. de Santana, E. C. B. Cambui, J. Ferreira-Ferreira, M. Oliveira, L. d. S. Barbosa, and C. L. Cordeiro, "Fire Detection with Deep Learning: A Comprehensive Review," *\*Land\**, vol. 13, no. 10, Art. no. 1696, Oct. 2024, doi: 10.3390/land13101696.
- ★ J. Pincott, P. W. Tien, S. Wei, and J. K. Calautit, "Development and evaluation of a vision-based transfer learning approach for indoor fire and smoke detection," *\*Building Services Engineering Research and Technology\**, vol. 43, no. 3, pp. 319-332, May 2022, doi: 10.1177/01436244221089445.
- ★ S. Geetha, C. S. Abhishek, and C. S. Akshayanat, "Machine Vision Based Fire Detection Techniques: A Survey," *\*Fire Technology\**, vol. 57, no. 3, pp. 591-623, May 2021, doi: 10.1007/s10694-020-01064-z.
- ★ L. He, Y. Zhou, L. Liu, Y. Zhang, and J. Ma, "Research and application of deep learning object detection methods for forest fire smoke recognition," *\*Sci. Rep.\**, May 2025, doi: 10.1038/s41598-025-98086-w.
- ★ R. Zhao et al., "Fire Detection Using Color and Texture Features with Support Vector Machines," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 06, p. 1355004, 2013.
- ★ A. Celik et al., "SVM based fire detection using color, texture and dynamic features," *Fire Safety Journal*, vol. 45, no. 4, pp. 253-258, 2010.



- ★ S. Sharma et al., "Comparison of Machine Learning Algorithms for Image-Based Fire Detection," in *2020 International Conference on Computing and Information Technology (ICCIT)*, 2020, pp. 1-6.
- ★ T. Chen et al., "Boosting based real-time forest fire detection," *Safety Science*, vol. 52, pp. 183-189, 2013.
- ★ Z. Zhang et al., "Image-based fire detection using MLP and RBF neural networks," in *2012 International Conference on Computer Science and Electronics Engineering*, 2012, vol. 2, pp. 450-453.
- ★ J. Wang et al., "An effective method for fire detection based on feature selection and SVM," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2018, pp. 547-552.
- ★ J. Kim and S. Lee, "A Custom Convolutional Neural Network for Real-time Fire Detection in Images," *Journal of Imaging Science and Technology*, vol. 64, no. 2, pp. 20501-1-20501-8, 2020.
- ★ L. Zhang et al., "A Lightweight CNN for Image-Based Fire Detection," in *2021 International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 4242-4250.

İnternetten konu araştırması için kullandığım kaynaklar (hiçbir yazıyı dümdüz almadım veya çevirerek buraya almadım, sadece yorum yapabilmek için kullandım)

- ★ <https://paperswithcode.com/methods/category/convolutional-neural-networks>
- ★ <https://paperswithcode.com/methods/category/pooling-operation>
- ★ <https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow/>
- ★ <https://www.geeksforgeeks.org/support-vector-machine-algorithm/>
- ★ <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>
- ★ <https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/>
- ★ <https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning>
- ★ <https://www.freecodecamp.org/news/build-a-multilayer-perceptron-with-examples-and-python-code/>
- ★ <https://odsc.medium.com/building-a-custom-convolutional-neural-network-in-keras-48171163aa7f>
- ★ <https://www.digitalocean.com/community/tutorials/writing-cnns-from-scratch-in-pytorch>
- ★ <https://aistudio.google.com/>

- ★ <https://milvus.io/ai-quick-reference/what-is-the-difference-between-hog-and-lbp>
- ★ <https://medium.com/swlh/human-skin-color-classification-using-the-threshold-classifier-rgb-ycbcr-hsv-python-code-d34d51febd8>
- ★ [https://www.academia.edu/65144103/Fire\\_Recognition\\_Using\\_RGB\\_and\\_Ycbcr\\_Color\\_Space](https://www.academia.edu/65144103/Fire_Recognition_Using_RGB_and_Ycbcr_Color_Space)
- ★ <https://ijcrt.org/papers/IJCRT1892785.pdf> (üzerinden literatür analizine gidilen bir materyal değil)
- ★ <https://datascience.stackexchange.com/questions/5706/what-is-the-dying-relu-problem-in-neural-networks>
- ★ <https://towardsdatascience.com/the-dying-relu-problem-clearly-explained-42d0c54e0d24/>
- ★ <https://www.geeksforgeeks.org/deep-learning/optimization-rule-in-deep-neural-networks/>
- ★ <https://www.ibm.com/think/topics/gradient-boosting>