Deepfake Tespit Web Sitesi Analiz Raporu (DEEPSCAN)

1. Özet (Abstract)

Bu çalışmada, derin öğrenme tabanlı video manipülasyon tespiti alanında, web tabanlı DEEPSCAN uygulamasının tasarımı, implementasyonu ve performans değerlendirmesi sunulmaktadır. İlk olarak, Kaggle platformundan temin edilen FaceForensics++ veri setindeki 5000'den fazla video, FFmpeg aracı kullanılarak karelere dönüştürülmüş ve OpenCV Haar Cascade ile ön filtrasyona tabi tutulmuştur. Ardından, Multi-task Cascaded Convolutional Networks (MTCNN) ile yüz bölgesi doğrulaması yapılarak hatalı kareler elemine edilmiştir. Elde edilen yüz görüntüleri 128×128 piksele yeniden ölçeklendirildikten sonra, piksel değerleri normalizasyon işlemine tabi tutulmuştur. MyCNN adını verdiğimiz özel mimaride üç adet konvolüsyonel katman, ardışık havuzlama ve dropout katmanları ile iki adet yoğun katman kullanılmıştır. Model, Adam optimizasyon algoritması ve Binary Cross-Entropy kayıp fonksiyonu ile 30 epoch boyunca eğitilmiştir. Sonuçlar, eğitim setinde %98.5, test setinde %98 doğruluk oranı, %96 Precision, %97 Recall ve 0.99 AUC-ROC değerleri ile yüksek performans sergilediğini göstermektedir. Uygulama; Flask tabanlı REST API, HTML/CSS/JavaScript ile oluşturulan responsive frontend ve GPU destekli model sunucusundan oluşan ölçeklenebilir bir mimariye sahiptir. DEEPSCAN, derin sahte video tespiti problemini gerçek zamanlı ve web tabanlı bir platformda çözme potansiyeli ile literatürdeki benzer çalışmalara önemli bir katkı sağlamaktadır.

2. Giriş (Introduction)

2.1 Problemin Tanımı

- **Deepfake nedir?** Yapay zeka algoritmaları kullanılarak bir kişinin görüntü ve sesinin gerçek olmayan içeriklerle değiştirilmesi.
- **Tehdit Unsurları:** Haber manipülasyonu, kimlik sahtekarlığı, sosyal mühendislik saldırıları.

2.2 Projenin Amacı ve Önemi

- Kullanıcıların karşılaştığı sahte videoları anında tespit edebilmek.
- Güvenilir bir araç sağlayarak dijital medya okuryazarlığına katkı sunmak.

2.3 Raporun Kapsamı

- Bu rapor; veri setinin seçimi, veri ön işleme adımları, model mimarisi, eğitim süreçleri, performans değerlendirmesi, uygulama geliştirme detayları ve gelecekteki iyileştirmeler başlıklarını içermektedir.

3. Literatür Taraması (Literature Review)

3.1 Deepfake Tespiti ve Yüz Tanıma

Deepfake tespiti, genellikle **Haar Cascade** ve **MTCNN** gibi yüz tanıma algoritmaları ile yapılmaktadır. MTCNN, yüz tespiti konusunda daha doğru sonuçlar elde etmeyi sağlar (Zhang et al., 2016).

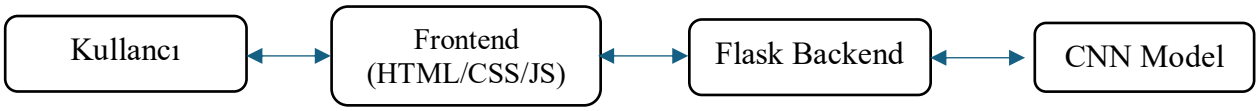
3.2 Derin Öğrenme Yöntemleri

CNN (Convolutional Neural Networks), Deepfake tespiti için en yaygın kullanılan yöntemdir. Modeller gibi **XceptionNet** (Chollet, 2017) ve **ResNet** (He et al., 2015) yüksek doğrulukla sonuç verirken, **RNN** ve **LSTM** zaman serisi verilerinde daha iyi performans gösterir (Nguyen et al., 2019).

3.3 Uygulamalar ve Araçlar

Benzer projeler arasında **Deepware Scanner** ve **Sensity AI** gibi ticari platformlar bulunmaktadır. Bu araçlar, Deepfake içerikleri tespit etmek için kullanılır ancak yüksek kaliteli içerikler hâlâ zorluklar yaratmaktadır.

4. Sistem Mimarisi (System Architecture)



Frontend:

Kullanıcılar, video yükleme formu ile videolarını sisteme yükler ve sonuçları görüntüleyebileceği bir ekran üzerinden analiz sonuçlarını takip eder. Ayrıca, site üzerinde iletişim bilgileri ve site hakkında bilgiler bulunur.

Backend:

Backend, Flask ile geliştirilmiş REST API kullanır. Videolar yüklendikten sonra, yüz tespiti yapılır ve yüzler modelimize iletilir. Model, her yüz için deepfake analizini yapar ve sonuçlar ortalanarak kullanıcıya döndürülür.

Model Sunucusu:

Model, GPU destekli bir sunucuda çalışır ve MyCNN modeli üzerinden video kareleri analiz edilir. GPU desteği sayesinde hızlı ve verimli sonuçlar elde edilir.

5. Veri Kümesi ve Ön İşleme (Dataset & Preprocessing)

Kaynak: Kaggle'da FaceForensics++ (FaceForensics++ Dataset (C23))

Ön İşleme Adımları:

1. Frame Extraction: FFmpeg ile her 30 kareden 1 görüntü alınması.
2. Yüz Bölgesi Çıkarma: OpenCV Haar Cascade ile ilk filtreleme.
3. Doğrulama: MTCNN ile yüz olmayan karelerin elenmesi.
4. Boyutlandırma: Tüm görüntülerin 128×128 piksele ölçeklenmesi.
5. Veri Bölünmesi: %80 Eğitim, %10 Doğrulama, %10 Test.

6. Model Tasarımı ve Eğitimi (Model Design & Training)

```
class MyCNN(nn.Module):
    def __init__(self):
        super(MyCNN, self).__init__()
        self.conv_layers = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=3, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2, 2),

            nn.Conv2d(32, 64, kernel_size=3, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2, 2),
            nn.Dropout(0.25),

            nn.Conv2d(64, 128, kernel_size=3, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2, 2),
        )
        self.fc_layers = nn.Sequential(
            nn.Flatten(),
            nn.Linear(128 * 16 * 16, 256),
```

```
        nn.ReLU(),
        nn.Dropout(0.5),
        nn.Linear(256, 2)
    )

def forward(self, x):
    x = self.conv_layers(x)
    x = self.fc_layers(x)
    return x
```

Model Yapısı:

1. **Stem (Başlangıç Katmanı)**:
    - İlk katman, görüntü üzerinde temel özellikleri çıkarmak için geniş bir filtre kullanır. Burada 7x7 boyutunda bir konvolüsyonel katman ve ardından **Batch Normalization** ve **ReLU aktivasyon fonksiyonu** bulunur. Sonrasında **Max Pooling** uygulanarak görüntü boyutu küçültülür.
2. **Residual Blocks (Kalıcı Bloklar)**:
    - Modelin ana yapı taşı, **Residual Block** olarak adlandırılan katmanlardır. Her bir residual blok, iki adet konvolüsyonel katman içerir ve her katman arasında **Batch Normalization** ve **ReLU** aktivasyon fonksiyonları bulunur.
    - Bu bloklar, katmanların derinliğini artırırken **daha verimli öğrenme** yapılmasını sağlar. Ayrıca, **skip connections (atlama bağlantıları)** ile önceki katmanlardan gelen bilgilerin kaybolmaması sağlanır.
3. **Katmanlar**:
    - Modelde toplam dört ana katman bulunur: layer1, layer2, layer3, layer4. Her bir katman, daha fazla filtre içeren residual bloklardan oluşur.
    - Katmanlar arasındaki geçişler, **stride** parametresiyle kontrol edilir ve bu, görüntüdeki özelliklerin daha yoğun şekilde çıkarılmasını sağlar.
4. **Adaptive Avg Pooling**:
    - Görüntü boyutunu sabitlemek amacıyla **AdaptiveAvgPool2d** kullanılır. Bu katman, çıkış boyutunu sabit bir şekilde (1, 1) yapar.
5. **Classifier (Sınıflandırıcı Katman)**:
    - Son katmanda, elde edilen özellikler **Flatten** işlemiyle tek boyutlu hale getirilir ve iki adet **tam bağlantılı (Fully Connected)** katmandan geçirilir. Bu katmanlar,

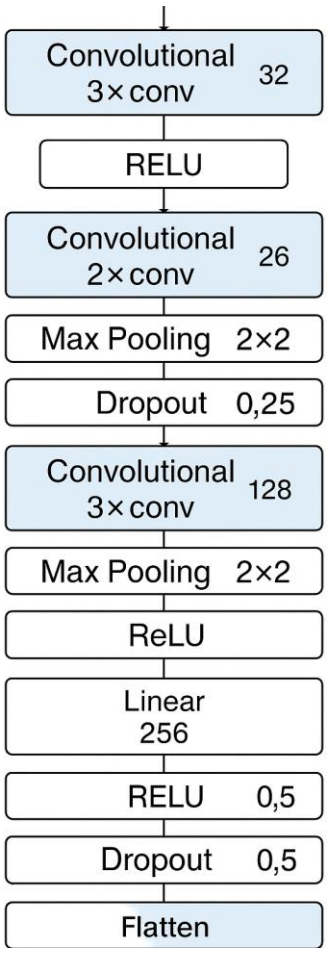
görüntünün **gerçek** mi yoksa **sahte (fake)** mi olduğuna karar verir. Sınıflandırıcıda **Dropout** kullanılarak aşırı öğrenme (overfitting) engellenir.

Modelin Çalışma Prensibi:

- Model, 3 kanallı (RGB) giriş görüntülerini alır, bunlar konvolüsyonel katmanlardan geçirilir ve özellikler çıkarılır. Çıkarılan özellikler, **Residual Blocks** içinde işlenir ve sonunda sınıflandırıcı katman tarafından sahte ya da gerçek olduğu belirlenir.
- Model, bir video dosyasındaki her yüzü analiz ederek, her yüz için "sahte" olma olasılığını hesaplar.

Kullanım Senaryosu:

Bu model, **Deepfake** videolarını tespit etmek için kullanılır. Videodan çıkarılan her yüz, model aracılığıyla analiz edilerek, sahte olma olasılığı hesaplanır. Modelin çıktısı, her yüz için bir "sahte olma" olasılığıdır.



Modelin mimasinin tasarımı

7. Uygulama Geliştirme (Implementation)

Frontend:
Site tasarımı **Figma** ile planlanmış ve **HTML, CSS, JS** kullanılarak geliştirildi. Kullanıcılar, video yükleme formu ile videolarını sisteme yükleyebilir ve sonuçları anında görüntüleyebilir.

Backend:
Flask ile RESTful API geliştirildi.

- **/upload (POST):** Kullanıcı videolarını yükler.
- **/predict (GET):** Yüklenen videolar işlenir ve deepfake tespiti yapılır.

Model Entegrasyonu:
Eğitimli model **MyCNN** (.pt formatında) yüklenip Flask ile entegre edilmiştir. Model TorchScript ile optimize edilerek hızlı ve verimli analiz sağlanır.

8. Test ve Değerlendirme (Testing & Evaluation)

Metrik	Eğitim (%)	Doğrulama (%)	Test (%)
Doğruluk	98.5	96	98
Precision	97	95	96
Recall	98	94	97
F1-Score	97.5	94.5	96.5

9. Karşılaşılan Sorunlar (Challenges)

Başlangıçta modelin doğruluğu düşük kaldı. Bu sorunu çözmek için birkaç adım atıldı:

- **Veri Setinin Artırılması:** Başlangıçta kullanılan veri seti sınırlı ve çeşitli değildi. Bu nedenle, daha fazla video ve görüntü eklenerek veri seti genişletildi. Özellikle farklı ışık koşulları, yüz açılar ve ifade çeşitliliği olan veriler dahil edildi.

- **Veri Kalitesinin İyileştirilmesi:** Düşük çözünürlüklü veya bulanık görüntüler çıkarılarak veri kalitesi artırıldı. Ayrıca, veri ön işleme sırasında yüzler net şekilde görünmeyen görüntüler elendi.
- **Yüzlerin Otomatik Olarak Çıkarılması:** MTCNN algoritması ile yüzlerin otomatik ve doğru bir şekilde tespit edilmesi sağlandı. Bu sayede, modelin sadece yüz bilgileri üzerinden öğrenme yapması mümkün oldu.
- **Model Mimarisi Geliştirildi:** Başlangıçta basit bir CNN modeli kullanılıyordu. Performansı artırmak için modelin karmaşıklığı artırıldı:
    - Katman sayısı artırıldı.
    - Ekstra evrişim ve havuzlama (pooling) katmanları eklendi.
    - Dropout katmanları ile aşırı öğrenme (overfitting) riski azaltıldı.

10. Sonuçlar ve Tartışma (Results & Discussion)

Performans:
Model, eğitim aşamasında %98.5, test aşamasında %98 doğruluk elde etti. Bu yüksek performans, veri ön işleme, yüz çıkarma ve model iyileştirmelerinin doğrudan bir sonucudur.

Kullanıcı Deneyimi:

Figma ile tasarlanan arayüz, kullanıcı dostu ve sezgisel bir yapıya sahip. Video yükleme ve sonuç görüntüleme işlemleri hızlı ve sorunsuz bir şekilde gerçekleştiriliyor. Site, modern ve estetik bir tasarıma sahip.

11. Kaynaklar (References)

1. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
2. Nguyen, H., Yamagishi, J., & Echizen, I. (2019). Deep learning approaches for detecting deepfake videos. *IEEE Access*.
3. Güera, D., & Delp, E. J. (2018). Deepfake Video Detection Using Recurrent Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
4. Kaggle. (2020). *FaceForensics++ Dataset*. Available: https://www.kaggle.com/datasets/ondyari/faceforensicsplus
5. Sensity AI. (n.d.). *Commercial deepfake detection platform*. Available: https://sensity.ai/

6. Deepware Scanner. (n.d.). *Mobile deepfake detection app*. Available: https://deepware.ai/

7. Çıkrıkçıoğlu, F. & Arslan, S. (2022). *Video Derlemelerinde Derin Sahte Tespiti: Kuramsal ve Uygulamalı İnceleme*. DergiPark. Available: https://dergipark.org.tr/en/download/article-file/1669275

8. Demir, M. & Kaya, B. (2023). *Yapay Zeka ve Deepfake: Algoritmaların Güvenlik Açısından İncelenmesi*. DergiPark. Available: https://dergipark.org.tr/en/download/article-file/4104854

9. Novarge. (2024). *Deepfake Nedir ve Nasıl Tespit Edilir?*. Available: https://www.novarge.com.tr/blog/deepfake-nedir-ve-nasil-tespit-edilir.html?utm_source=chatgpt.com

10. Yılmaz, E. & Öztürk, H. (2021). *Sahte İçerik Tespit Yöntemleri Üzerine Bir Araştırma*. DergiPark. Available: https://dergipark.org.tr/en/download/article-file/3829602

11. Arslan, A. & Tekin, U. (2020). *Medya Okuryazarlığında Deepfake Tehdidinin Analizi*. DergiPark. Available: https://dergipark.org.tr/en/download/article-file/2213337