

# Academic Report Draft

## 1. Problem ve Veri

---

**Problem Tanımı:** Bu proje, video zaman serisi verilerini kullanarak bir sonraki kareyi tahmin etme (Next-Frame Prediction) problemini ele almaktadır. Amaç, verilen ardışık video karelerinden (0-18. kareler) oluşan bir diziyi girdi olarak alıp, bu dizinin bir adım sonrası (1-19. kareler) veya 20. kareyi tahmin edebilen bir derin öğrenme modeli geliştirmektir. Bu problem, meteorolojik tahminler, otonom sürüş ve video sıkıştırma gibi alanlarda kritik öneme sahiptir.

**Veri Kaynağı:** Projede, standart bir kıyaslama veri seti olan **Moving MNIST** veri seti kullanılmıştır. Bu veri seti, `tensorflow_datasets` kütüphanesi aracılığıyla elde edilmiştir.

**Zaman Aralığı:** Veri seti sentetik olduğu için belirli bir gerçek dünya zaman aralığı (yıl, ay vb.) yoktur. Ancak her bir veri örneği, ardışık **20 zaman adımından (kare)** oluşan bir video kesitidir.

**Gözlem Şekli:** Veriler video formatındadır ve ardışık kareler (frames) şeklinde düzenlenmiştir. Her bir kare, zaman serisindeki bir gözlem anını temsil eder.

**Kullanılan Değişkenler:** Veri setindeki temel değişken, piksel yoğunluk değerleridir. \* **Girdi:** 64x64 boyutunda, tek kanallı (siyah-beyaz) görüntüler. \* **Değer Aralığı:** 0-255 arasındakiler (ön işlemede 0-1 aralığına normalize edilmiştir).

## 2. Veri Ön İşleme

---

Veri seti model eğitime uygun hale getirilmek için aşağıdaki adımlardan geçirilmiştir:

**Ölçekleme (Scaling):** Orijinal veri setindeki 0-255 aralığındaki piksel değerleri, [0, 1] aralığına normalize edilmiştir. Bu işlem, sınır aklarının daha hızlı ve kararlı yakınsamasının sağlamak ve aktivasyon fonksiyonlarının (Relu, Sigmoid) verimli çalışması için yapılmıştır.  $x_{norm} = x / 255.0$

### Dönüşümler:

- Boyutlandırma:** Görüntüler 64x64 çözünürlüğünde işlenmiştir. Model girdisi (Batch, Time, Height, Width, Channel) formatına uygun olarak (19, 64, 64, 1) şeklinde yeniden boyutlandırılmıştır.
- Girdi-Çıktı Ayrımı:** 20 karelik video dizileri şu şekilde ayrılmıştır:
  - Girdi (X):** İlk 19 kare (t=0'dan t=18'e kadar).
  - Hedef (Y):** Sonraki 19 kare (t=1'den t=19'a kadar). Bu yapı, modelin "Sequence-to-Sequence" (Diziden Diziye) öğrenme yapmasını desteklemektedir.

**Eksik Veri İşlemleri:** Moving MNIST sentetik ve tam bir veri seti olduğu için eksik veri bulunmamaktadır. Bu nedenle eksik veri doldurma işlemine ihtiyaç duyulmamıştır.

### 3. Kullanılan Yöntem

---

Bu projede, spatiotemporal (uzay-zamansal) özellikleri öğrenmek için bir **ConvLSTM (Convolutional Long Short-Term Memory)** ağı tercih edilmiştir.

**Seçilen Model:** Model, konvolüsyonel sinir ağlarının (CNN) görsel özellik çıkarma yeteneği ile LSTM ağlarının zamansal bağımlılıkları öğrenme yeteneğini birleştiren **ConvLSTM2D** katmanlarından oluşmaktadır.

**Model Mimarisi:** 1. **Giriş Katmanı:** (19, 64, 64, 1) boyutunda dizi. 2. **ConvLSTM2D Katman 1:** 64 Filtre, 5x5 Kernel. Geniş alandaki hareketleri yakalar. 3. **ConvLSTM2D Katman 2:** 64 Filtre, 3x3 Kernel. Daha detaylı hareket özelliklerini öğrenir. 4. **ConvLSTM2D Katman 3:** 64 Filtre, 1x1 Kernel. Özellik haritalarını birleştirir ve derinlik katar. 5. **Çıkış Katmanı (Conv3D):** 3x3x3 Kernel, Sigmoid aktivasyon. Zamansal ve uzaysal düzeltme yaparak nihai görüntüyü [0, 1] aralığında üretir.

**Modelin Seçilme Nedeni:** Standart CNN'ler zamansal bilgiyi (hareket yönü, hızı) yakalayamazken, standart LSTM'ler görüntüdeki uzaysal ilişkileri (nesne şekli, konumu) kaybeder. ConvLSTM, hücre içindeki matris çarpımlarını konvolüsyon işlemiyle değiştirerek hem uzaysal yapıyı korur hem de zamansal değişimi modeller. Bu nedenle video tahmin problemleri için idealdir.

**Model Parametreleri:** \* **Optimizasyon Algoritması:** Adam (Adaptive Moment Estimation) \* **Öğrenme Oranı (Learning Rate):** 5e-4 \* **Batch Size:** 8 \* **Epoch Sayısı:** 50 \* **Kayıp Fonksiyonu (Loss Function):** Toplam Kayıp = BCE + Perceptual + GDL \* **BCE (Binary Cross Entropy - 10.0):** Piksel bazında doğruluk sınırları. \* **Perceptual Loss (0.1):** VGG16 ağı kullanılarak algısal benzerliği artırır (görüntülerin insan gözüne doğal

görünmesi). \* **GDL (Gradient Difference Loss - 0.1)**: Görüntüdeki keskinlik ve kenar detayların korunmasını sağlar.

## 4. Uygulama ve Sonular

---

**Eğitim / Test Ayrımı:** Veri seti, modelin genelleme yeteneğini ölçmek amacıyla eğitim ve doğrulama (validation) setlerine ayrılmıştır. \* **Doğrulama Seti:** İlk 1000 örnek. \* **Eğitim Seti:** Geriye kalan veriler (karşılaştırılarak kullanılmıştır).

**Hata Ölçütleri:** Modelin başarımları aşağıdaki metriklerle değerlendirilmiştir: \* **PSNR (Peak Signal-to-Noise Ratio):** Görüntü kalitesini ölçer. Yüksek olması iyidir. \* **SSIM (Structural Similarity Index):** Yapısal benzerliği ölçer (0-1 arası). 1'e yakın olması iyidir. \* **MSE (Mean Squared Error):** Hata karesi ortalaması. Düşük olması iyidir. \* **MAE (Mean Absolute Error):** Ortalama mutlak hata.

**Sonuçlar:** Yapılan testler sonucunda elde edilen ortalama değerler (Bkz: `evaluate.py` çıktısı): \* **Ortalama PSNR:** X.XXXX (Örnek: 25.00+) \* **Ortalama SSIM:** 0.XXXX (Örnek: 0.85+) \* **Ortalama MSE:** 0.XXXX \* **Ortalama MAE:** 0.XXXX

**Tahmin Grafikleri:** Elde edilen sonuç görselleri (`results/evaluation_results.png`) rapor ekinde sunulmuştur. Bu görsellerde sırasıyla "Gerçek Görüntü (Ground Truth)" ve "Model Tahmini (Prediction)" yan yana karşılaştırılarak modelin hareketli rakamları takip etme ve bütünlüğünü koruma başarısı gösterilmiştir. Ayrıca eğitim süreci boyunca kaybın değişimi `results/loss_curve.png` grafiğinde verilmiştir.