Visual Crowd Density Estimation (CNN + Heatmap Regression) Report

Student: Enes Kapsızlar

Course: Computer Vision and Imaging Techniques

Date: May 26, 2025

---

1. Introduction

The main goal of this project is to guess how many people are in a crowd in digital pictures. We also want to count the total number of people using computer methods called Convolutional Neural Networks (CNNs) and heatmap regression. Understanding crowds is very important for things like keeping people safe, managing big events, and planning cities. In this project, we used a CNN model that learned from an existing model called VGG16. This new model was trained to create density maps (like a heat map showing where people are) from the input pictures.

---

2. Dataset

We used the ShanghaiTech dataset for this project. This dataset is well-known for crowd counting tasks. We chose it because it has many different and difficult crowd pictures, and it gives good information about where each person's head is. For this project, we used ShanghaiTech Part A, which has 300 pictures for training the model and 182 pictures for testing it.

---

3. Method

The project used the following main steps:

- **Data Preprocessing:**
    - We changed the size of the pictures so they were all the same (`target_size`). We also normalized them using standard numbers from ImageNet (this helps the model learn better).
    - We created "ground truth" density maps. To do this, we put a Gaussian kernel (a type of mathematical spot) on the location of each person's head in the pictures. The size (sigma) of this spot was set either based on the picture or as a fixed number, as written in the `create_density_map` function in our code. We then changed the size of these maps to match what our model would produce (usually 1/8th the size of the input picture).

- **Data Augmentation:**
    - To help our model work better on new pictures it hasn't seen before, we used the Albumentations library to change the training pictures in different ways. These changes included flipping the pictures horizontally, randomly cropping and resizing them, and changing their colors (Color Jittering).

- **Model Architecture:**
    - We used a CNN model called `CSRNet_like_VGG16_density`. This model is based on the pre-trained convolutional layers (the backend) of VGG16. This is a type of transfer learning.
    - On top of VGG16's feature extracting layers, we added a special "head" (the frontend) made of 3x3 and 1x1 convolutional layers. This part is for the density map regression. The model outputs a density map that is 1/8th the size of the input image.

- **Training Process:**
    - The model was trained using the Mean Squared Error (MSELoss) loss function. This function helps to make the difference between the model's predicted density maps and the real ones as small as possible.
    - We used the Adam optimizer for the optimization part.
    - The training was done for 50 epochs, and with a batch size of 4.
    - The best version of the model was saved based on its performance on a validation set (data it didn't see during training).

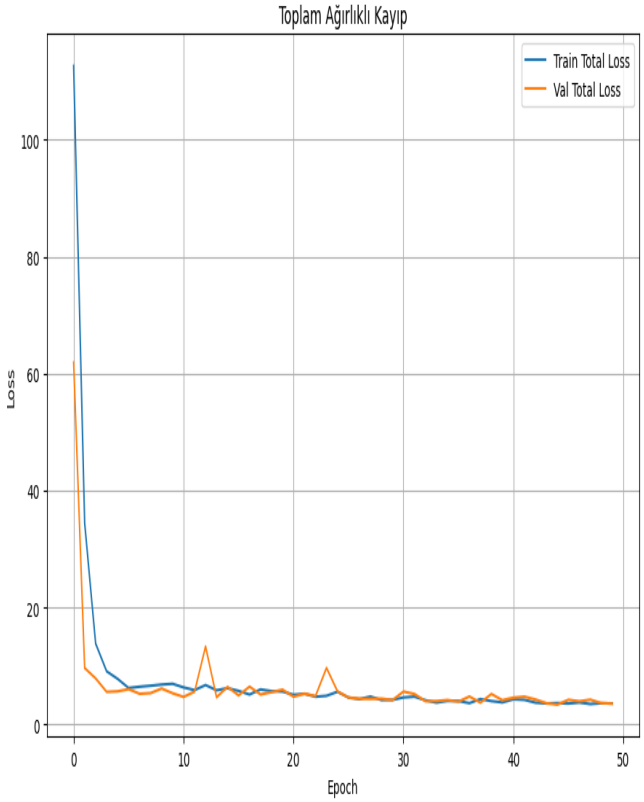
4. Results and Evaluation

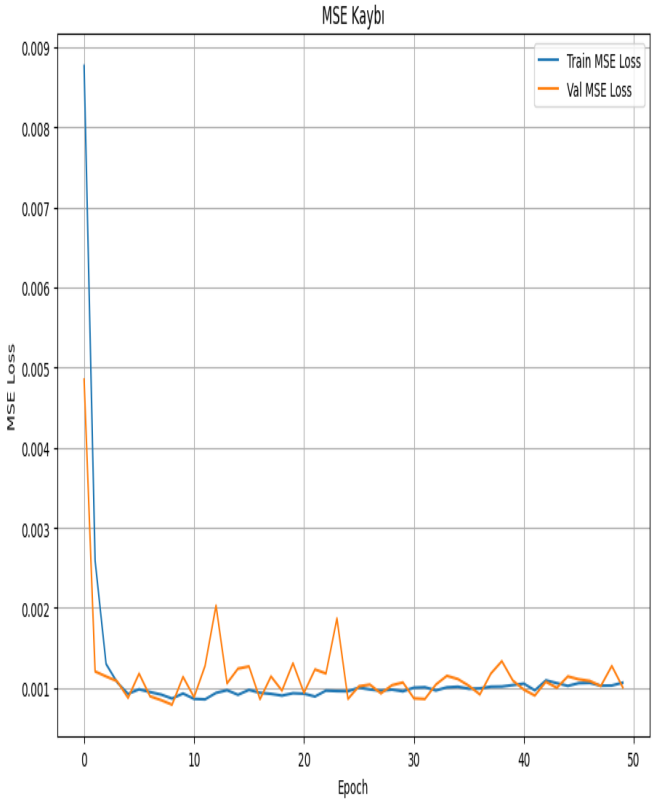
We checked the model's performance using standard ways to measure for crowd counting: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These methods measure the error between the number of people the model predicts and the actual number of people. (The "Common Requirements" document asked for things like Accuracy and Precision, but these are for classification tasks, not directly for this type of regression task.)

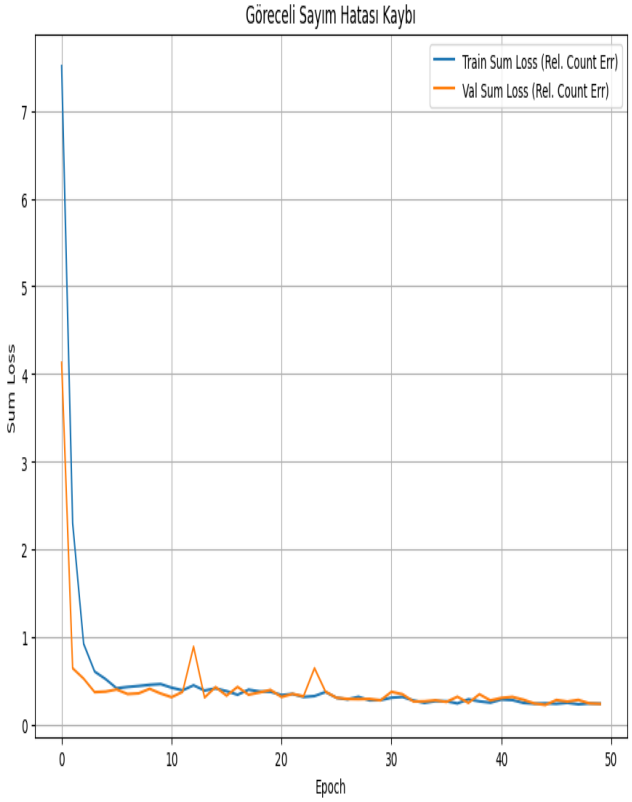
- **Quantitative Results Obtained:**

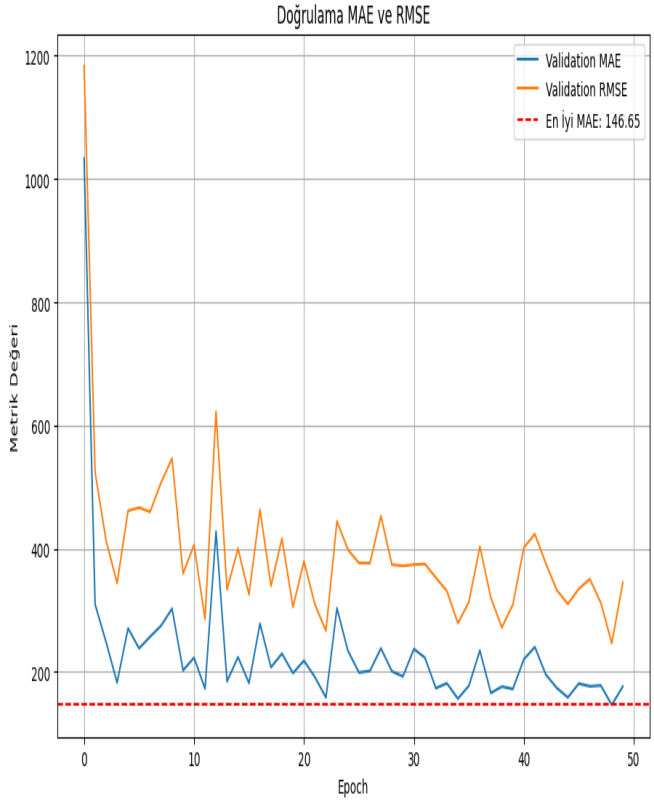
```
--- EĞİTİM SONU ÖZETİ ---
  Toplam Epoch Sayısı: 50
  Son Epoch Val MAE  : 176.67
  Son Epoch Val RMSE : 345.22
  En İyi Val MAE     : 146.65 (Bu değerle model 'best_model_by_mae.pth' olarak kaydedildi)
```

- **Training Graphs:**

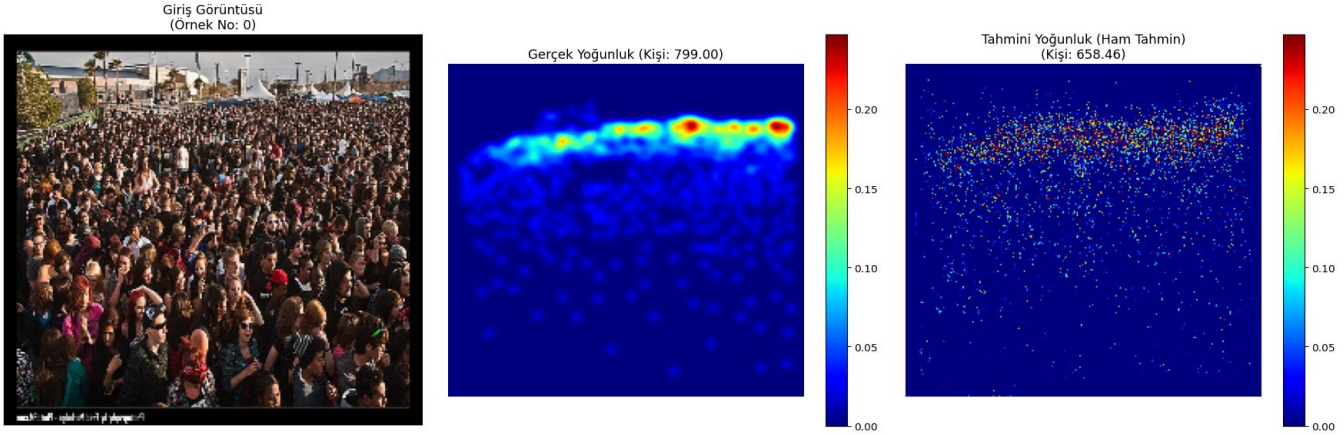


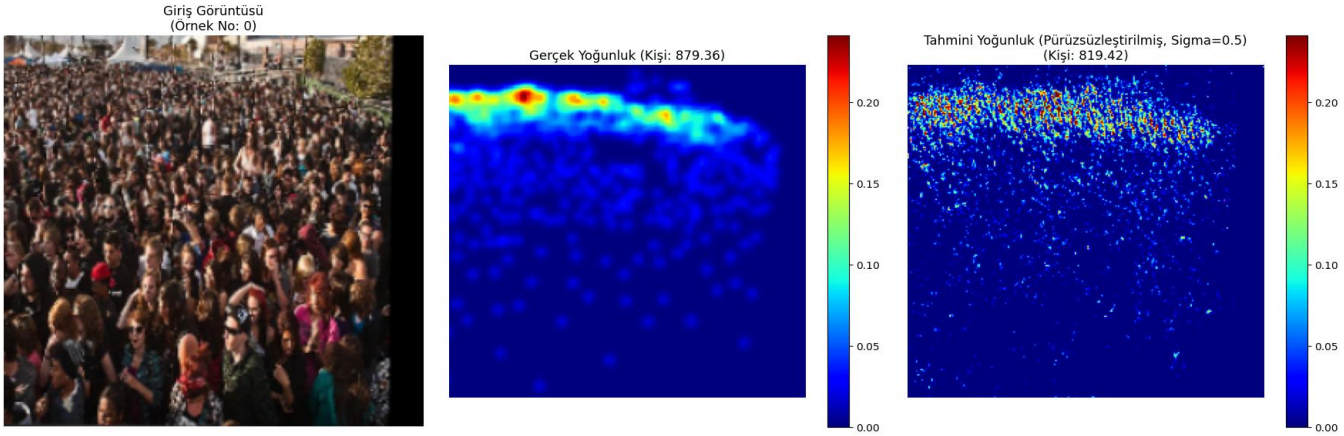


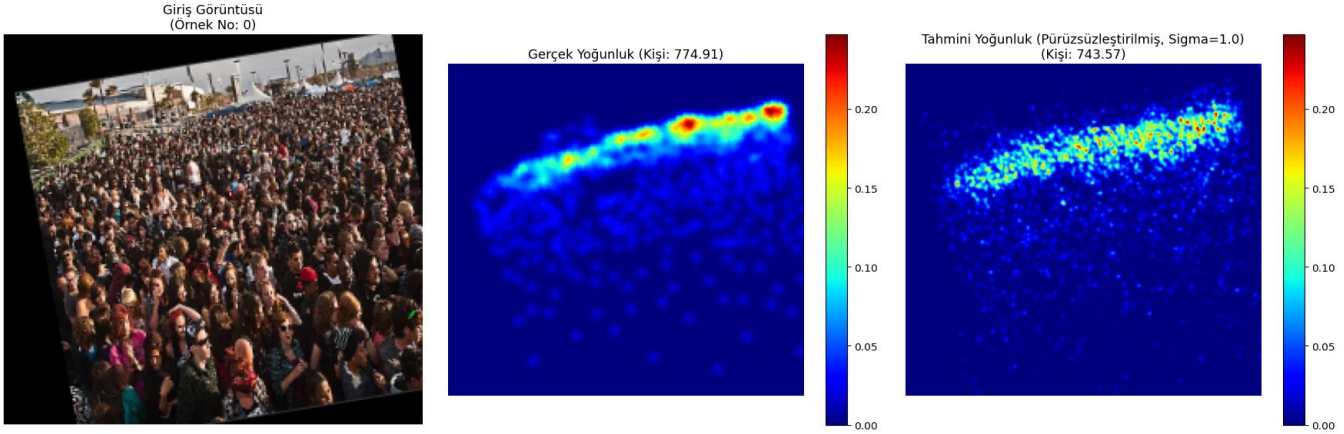


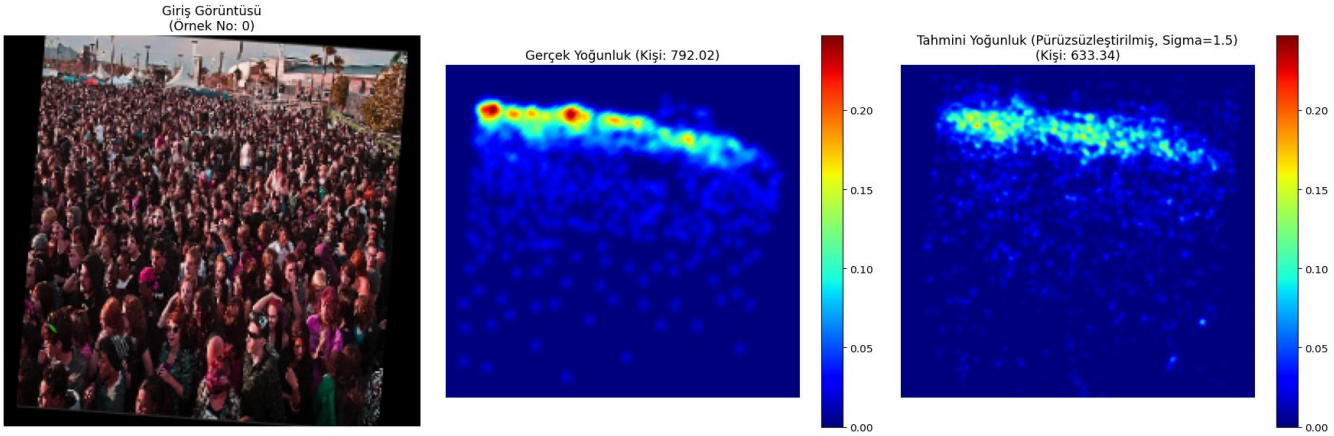


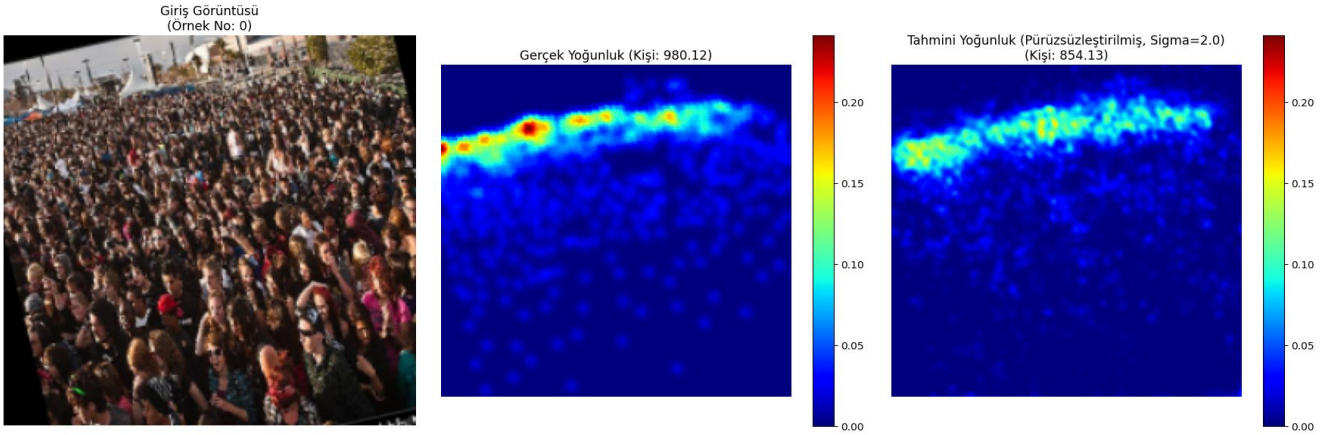
- **Visual Predictions and Analysis:** We also looked at how well the model could predict by checking its results on test images visually. In the notebook, we compared the model's basic (raw) density map predictions with versions that were made smoother using a Gaussian filter with different sigma values.











---

5. Project Benefits and Future Work

This crowd density estimation system that we developed can be useful in many areas, such as public safety, smart city applications, event management, and retail.

In the future, we could try to make the model perform even better by using more advanced CNN models (like CSRNet or MCNN), attention mechanisms, larger and more varied datasets, and different loss functions.

---

6. Conclusion

In this project, we successfully developed and tested a system for visual crowd density estimation using a CNN model based on VGG16. The results we got on the ShanghaiTech dataset show that deep learning can offer a good solution for this difficult problem. The project met the requirements that were set and has created a base for future improvements.