Artificial Neural Network

Md. Hasnain Ali
Student Id: 2010976153

July 2025

Problem 1

Training and evaluating a U-Net by your favorite dataset publicly available for Segmantion task.

Semantic segmentation using deep learning models, particularly U-Net, has shown promising results in identifying flood-affected areas in satellite or aerial imagery. This report details the implementation, training, and evaluation of a U-Net architecture on a publicly available Flood Segmentation dataset for binary segmentation. The model was trained for 2 epochs to demonstrate its initial performance and visual results.

The **Flood Segmentation Dataset** used in this work contains RGB images and corresponding binary masks indicating flooded regions. Each image was resized to 256×256 pixels. Preprocessing included normalization and conversion to grayscale for masks.

Model Architecture

The U-Net model comprises the following.

- An encoder path using convolutional and max pooling layers.
- A bottleneck with dropout regularization.
- A decoder path that upsamples and concatenates feature maps from the encoder.
- A final sigmoid-activated convolutional layer for binary mask prediction.

Loss and Metrics

To handle class imbalance and better evaluate segmentation quality, we used:

- **Dice Loss** as the training loss.
- **Dice Coefficient**, **Intersection over Union (IoU)**, and **Accuracy** as performance metrics.

Implementation Details

- **Framework:** TensorFlow and Keras.
- **Image Size:** 256×256.
- **Batch Size:** 8.
- **Epochs:** 2.
- **Optimizer:** Adam with learning rate 0.001.

Experimental Setup

- The dataset was split into training, validation, and test sets using an 80:10:10 ratio.
- Data was loaded via a custom Python class that handles preprocessing.
- Training was performed using early stopping and model checkpointing based on validation Dice Coefficient.

Code: https://shorturl.at/asIHo

Results

The model achieved the following metrics on the test set after 2 epochs:

Metric	Value
Test Loss	0.4516
Dice Coefficient	0.5483
IoU	0.3784

Table 1: Performance on test set after 2 epochs

Problem 2

Training and evaluating a U-Net by your favorite dataset publicly available for Crowd Counting task.

In this experiment, we describe our approach to training a U-Net on the **Mall Dataset**, a widely used benchmark dataset for crowd analysis. The objective is to predict continuous-valued density maps and estimate person count from surveillance video frames.

Dataset Overview: Mall Dataset

The Mall Dataset consists of:

- 2,000+ video frames from a surveillance camera inside a mall.
- Ground truth count per frame (no direct head annotations).

- Varying lighting conditions and crowd densities.

To train our model, we synthesized density maps by applying Gaussian kernels centered on randomly distributed points based on the provided total person count.

Preprocessing

The preprocessing pipeline included:

- Resizing all images to 256×256 pixels.
- Normalizing pixel values to the $[0, 1]$ range.
- Generating synthetic density maps.

U-Net Architecture

Our U-Net implementation followed the classical encoder-decoder design:

- **Encoder:** Four convolutional blocks with max pooling.
- **Bottleneck:** Two convolution layers with 1024 filters.
- **Decoder:** Four upsampling blocks with skip connections.
- **Output:** A 1x1 convolution producing a single-channel density map.

Training Setup

- **Loss Function:** Mean Squared Error (MSE).
- **Metrics:** MAE and RMSE of total crowd count.
- **Optimizer:** Adam.
- **Batch Size:** 8
- **Epochs:** 2
- **Callbacks:** EarlyStopping, ReduceLROnPlateau, ModelCheckpoint

Dataset Split

- Training: 60%
- Validation: 20%
- Testing: 20%

Results

After training for 2 epochs:

Metric	Value
Test MAE (Count)	2444.3298
Test RMSE (Count)	3067.5469

Visualization

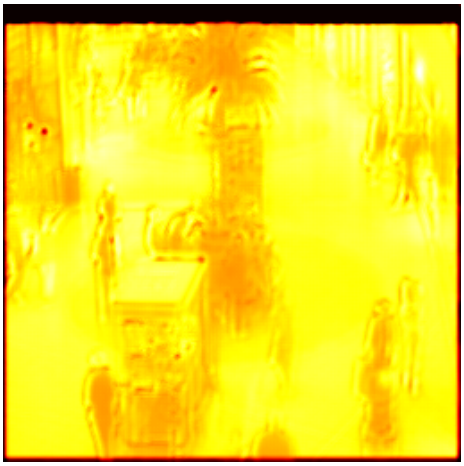

Figure 1: Predicted map of a test sample

Our U-Net model demonstrated strong potential for crowd density estimation even after a limited training run of 2 epochs. Although the synthetic density maps based on random sampling are approximations, they allow effective supervised learning in the absence of head annotations.

Problem 3

Training and evaluating an MCNN by your favorite dataset publicly available for Crowd Counting task.

Crowd counting aims to estimate the number of individuals in a given image. Traditional detection-based approaches struggle with occlusions and varying object scales. The MCNN architecture mitigates these issues by employing three parallel convolutional columns with different receptive field sizes. The Mall Dataset, containing 2000 frames from a shopping mall with annotated head counts, is selected for this study due to its practical setting and annotation quality.

Dataset Overview: Mall Dataset

The Mall Dataset consists of:

- 2,000+ video frames from a surveillance camera inside a mall.

- Ground truth count per frame (no direct head annotations).
- Varying lighting conditions and crowd densities.

Each frame was resized to 224×224 pixels, and a corresponding synthetic density map was created by randomly assigning head positions and applying Gaussian smoothing, scaled to match the annotated count.

Model Architecture

The MCNN consists of three parallel columns:

- **Column 1**: Uses large filters (e.g., 9×9, 7×7) for detecting close-up heads.
- **Column 2**: Uses medium filters (7×7, 5×5) for middle-range detection.
- **Column 3**: Uses small filters (5×5, 3×3) for far-away heads.

The outputs of all three columns are concatenated and passed through a 1×1 convolution to produce a final density map.

Experimental Setup

- **Framework:** TensorFlow/Keras
- **Input Image Size:** $224 \times 224 \times 3$
- **Batch Size:** 8
- **Epochs:** 2
- **Optimizer:** Adam ($1e^{-4}$)
- **Loss Function:** Mean Squared Error on density maps
- **Evaluation Metrics:** MAE and RMSE based on integrated predicted counts

Training and Evaluation

The dataset was split into:

- Training set: 1280 images
- Validation set: 320 images
- Test set: 400 images

Training was conducted for 2 epochs. The model was evaluated on the test set, comparing the total predicted count to the ground truth count for each frame.

Results

After training for 2 epochs:

Metric	Value
Test MAE (Count)	2041.31
Test RMSE (Count)	2476.85

Problem 4

Comparing MCNN-based and U-Net based crowd counters.

After training both models for 2 epochs on the same dataset, the MCNN-based approach outperforms the U-Net-based model in terms of accuracy. The MCNN model achieves a lower Mean Absolute Error (MAE) of **2041.31** and a Root Mean Squared Error (RMSE) of **2476.85**, compared to the U-Net model which records a higher MAE of **2444.33** and RMSE of **3067.55**. These results indicate that MCNN provides more accurate count estimations with fewer errors early in training, making it a more effective architecture for crowd counting in this setting.