**A logo of a university

Description automatically generatedUniversity of New Haven Tagliatela college of Engineering**

**A white background with black and white clouds

Description automatically generated**

**Master’s in data science**

Course: Deep Learning

**Title: Revising Deep Learning Methods in Parking Lot Occupancy Detection**

By

Trinadh Nandamuri

Medhini Barla

**Under the guidance of**,

Prof. Muhammad Aminul Islam, Ph.D

mislam@newhaven.edu

**Table Of Content**

[**1.** **Introduction** 3](#_Toc153610499)

[**2.** **Dataset Collection** 3](#_Toc153610500)

[**2.1.** **Data Transformations** 3](#_Toc153610501)

[**2.2.** **Data Augmentation** 4](#_Toc153610502)

[**2.3.** **Visualization** 4](#_Toc153610503)

[**3.** **Transfer Learning** 4](#_Toc153610504)

[**3.1.** **Loading Pretrained Model Parameters** 4](#_Toc153610505)

[**3.2.** **Modifying the Output Layer** 5](#_Toc153610506)

[**3.3.** **Fine-Tuning the Model** 5](#_Toc153610507)

[**3.4.** **Results and Discussion Metrics** 5](#_Toc153610508)

[**4.** **Mini Model** 6](#_Toc153610509)

[**4.1.** **Model Architecture** 6](#_Toc153610510)

[**4.2.** **Hypertuning Parameters** 7](#_Toc153610511)

[**4.3.** **Experiments and Results** 8](#_Toc153610512)

[**Deployment** 9](#_Toc153610513)

[**Conclusion** 9](#_Toc153610514)

[**References** 10](#_Toc153610515)

**Revising Deep Learning Methods in Parking Lot Occupancy Detection**

1. **Introduction**

Parking lot occupancy detection plays a pivotal role in enhancing contemporary traffic management systems. In response to the intricate challenges inherent in this domain, the current project is strategically designed to advance deep learning methodologies, aiming to bolster the efficiency of occupancy detection. The comprehensive nature of the project encompasses various crucial stages, including meticulous data collection, the creation of a specialized dataset, strategic partitioning, normalization of data to ensure consistent input formats, as well as judicious data augmentation techniques to enrich the diversity of the dataset. Furthermore, the project delves into the intricate process of model development, exploring architectures tailored to the unique characteristics of parking lot scenarios, and subsequently optimizing these models for robust performance. By addressing each of these key components, the project endeavors to contribute significantly to the advancement of parking lot occupancy detection systems, ultimately fostering more intelligent and responsive traffic management solutions.

1. **Dataset Collection**

To accurately represent real-world parking scenarios and challenges, a dataset of parking lot images has been collected. The dataset set is composed of 200 images 100 from occupied and 100 from empty car parking spaces. The dataset reflects diverse environmental conditions and occupancy states. The dataset is annotated to identify occupied and free parking spaces. The data used for this project is hosted at

DataSet:<https://drive.google.com/drive/folders/1pv1Zc0-0TLm3Q74BCaiuVm4vSJxipI3A?usp=drive_link>

**Data Preprocessing**

The dataset was preprocessed using resizing to 32x32 pixels and normalization based on ImageNet mean and standard deviation values. The custom dataset class (CustomDataset) was implemented to load and transform the images.

* 1. **Data Transformations**

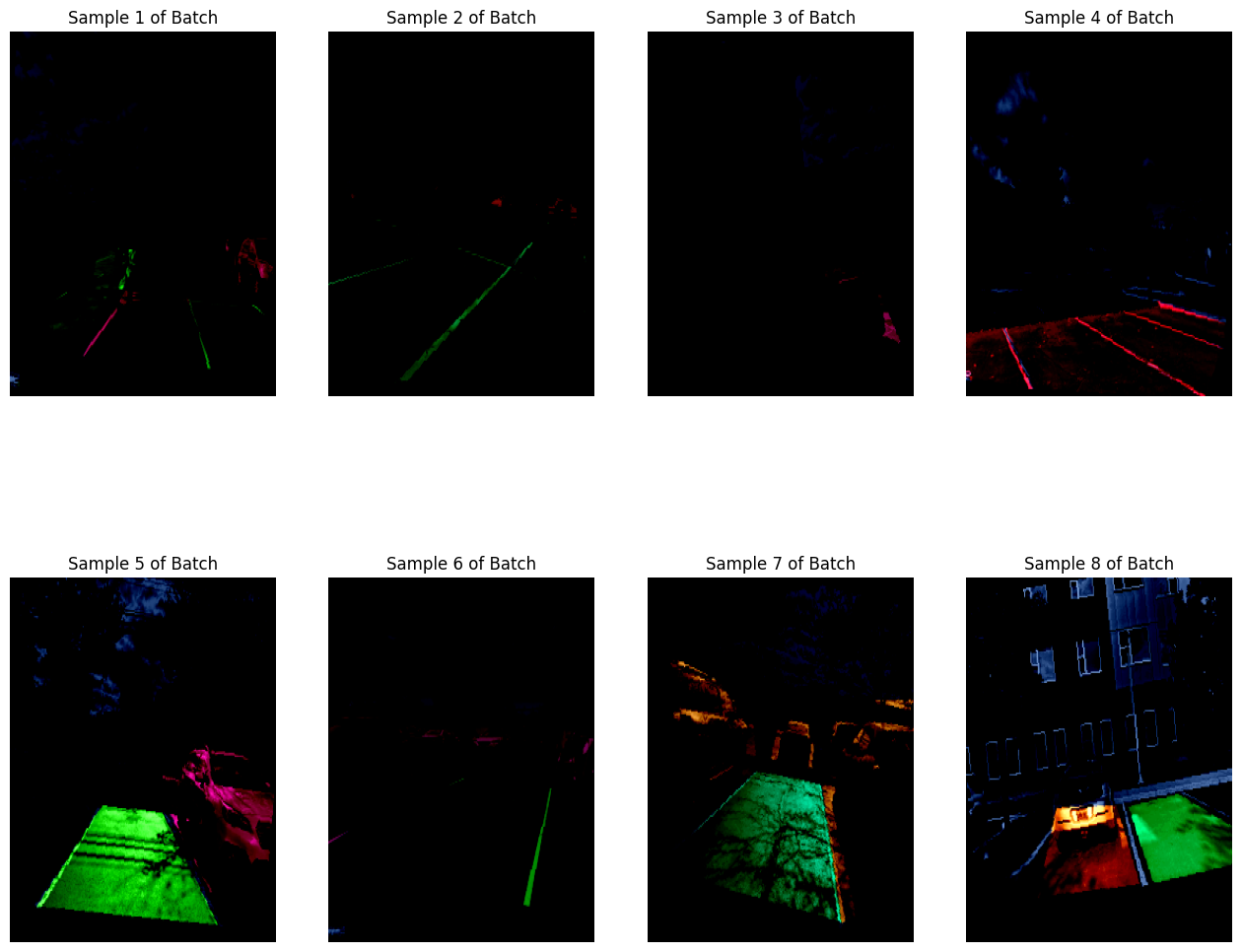
A set of data transformations has been applied to prepare the dataset for deep learning models. These transformations include:

1. Resizing images to a common size (256x256 pixels)
2. Converting images to PyTorch tensors
3. Applying data normalization with specified mean and standard deviation values
   1. **Data Augmentation**

Data augmentation is crucial to improve model robustness. Data augmentation techniques that have been implemented include:

1. Random horizontal flipping
2. Color jittering (brightness, contrast, saturation)
3. Random rotation
   1. **Visualization**

Understanding the data is vital. The first sample of each minibatch has been visualized to ensure that data preprocessing and augmentation produce the desired outcomes. This step helps identify any anomalies in the data.



1. **Transfer Learning**
   1. **Loading Pretrained Model Parameters**

The ResNet18 model was loaded with pre-trained weights, capturing the knowledge gained from training on a large dataset.

**

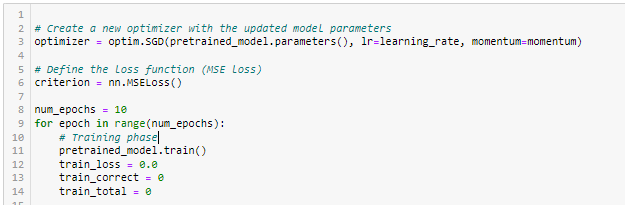
* 1. **Modifying the Output Layer**

The output layer of the model was modified to have a single neuron for regression.

**

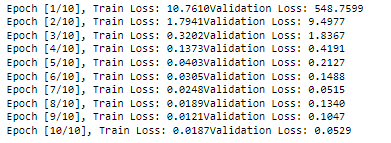
* 1. **Fine-Tuning the Model**

The model was fine-tuned using Mean Squared Error (MSE) loss and Stochastic Gradient Descent (SGD) optimizer. The training loop included both training and validation phases for hyperparameter tuning.



* 1. **Results and Discussion Metrics**

The model underwent fine-tuning over 10 epochs. The training and validation loss values for each epoch are as follows:

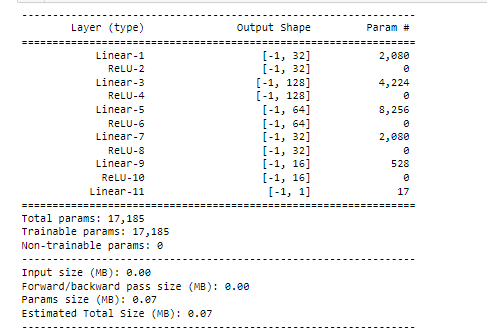


As seen in the graph below the training loss (blue line): The training loss is steadily decreasing throughout the epochs, which is a good sign. This means that the model is learning from the training data and improving its ability to make predictions. Validation loss: The validation loss initially decreases but then starts to increase again after epoch 4. This is a sign that the model is starting to overfit to the training data. In other words, the model is learning the training data too well, but it is not generalizing well to unseen data.



The model was evaluated on a test dataset, yielding a Mean Squared Error (MSE) of 0.0257

1. **Mini Model** 
   1. **Model Architecture**

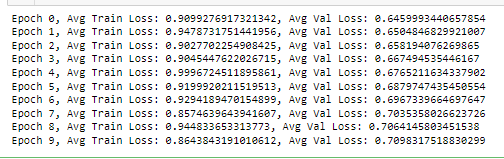


The presented model architecture consists of a series of fully connected (Linear) layers interspersed with Rectified Linear Unit (ReLU) activation functions, culminating in a single output layer. The input to the model is expected to be a vector, and each linear layer transforms the input into a different feature representation with varying dimensions. The architecture begins with an input layer connected to a hidden layer of 32 units, followed by a ReLU activation function. Subsequent layers include a hidden layer of 128 units, another ReLU activation, a layer with 64 units, another ReLU activation, a layer with 32 units, yet another ReLU activation, a layer with 16 units, and a final ReLU activation. The output layer, a single-unit linear layer, produces the final prediction. The model comprises a total of 17,185 parameters, all of which are trainable. This architecture suggests a hierarchical feature extraction process, progressively capturing complex patterns and representations in the data. The model's estimated total size is 0.07 MB, indicating a lightweight structure suitable for scenarios with limited computational resources.

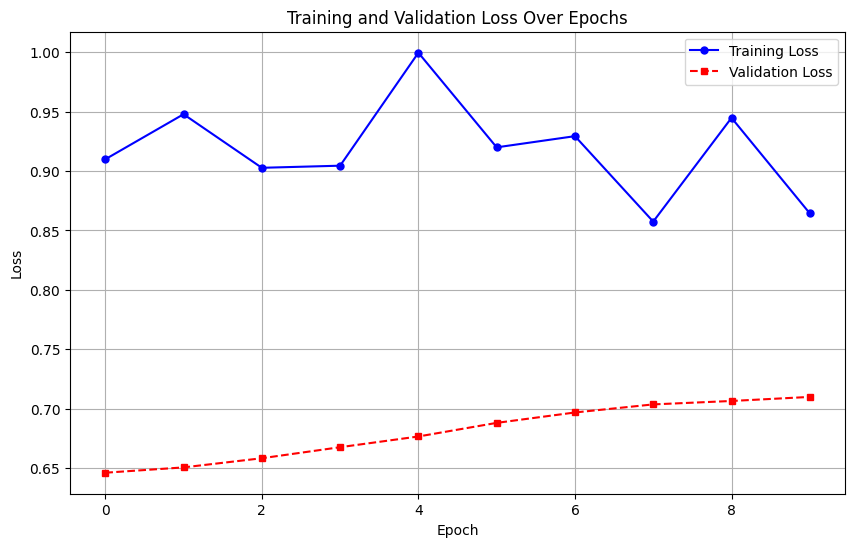
* 1. **Hypertuning Parameters**

The optimizer employed is Stochastic Gradient Descent (SGD) with the following parameters:

* lr=0.001: Learning rate, controlling the step size for parameter updates.
* momentum=0.9: Momentum term, incorporating a fraction (90%) of the previous gradient to accelerate convergence.
* weight\_decay=0.0001: Weight decay, a regularization term penalizing large weights to prevent overfitting.
  1. **Experiments and Results**

****

The training process is observed over ten epochs, where the average training loss and average validation loss are reported. In the initial epoch (Epoch 0), the average training loss is 0.91, and the average validation loss is 0.65. Subsequent epochs show fluctuations in both training and validation losses. Notably, in Epoch 4, there is a spike in the average training loss to 1.0, indicating a potential challenge in model convergence. However, this is followed by a gradual decrease in subsequent epochs. The model appears to struggle with minimizing the validation loss consistently, as it slightly increases over epochs. Epoch 7 shows a decrease in both training and validation losses, suggesting a temporary improvement, but this trend is not sustained in the final epochs. This is also evident in the graph below;

****

The reported average test loss is 1.07, providing an evaluation metric for the model's performance on unseen data. The test loss quantifies the disparity between the predicted outputs and the actual values in the test set, with a lower value indicating better model generalization.

# **Deployment**

Gradio is a user-friendly framework for creating interactive web interfaces for machine learning models. The Gradio interface for parking occupancy detection allows users to upload parking lot images and receive real-time predictions of occupancy status

Input : we have to upload parking lot images which is not annotated.

Output: model train the image and it shows which is parking and occupied.

A screenshot of a computer

Description automatically generated

# **Conclusion**

This project addresses parking lot occupancy detection using deep learning. The report has described dataset collection, dataset creation, partitioning, data preprocessing, data augmentation, model development, and the optimization process. Visualizing the data aids in ensuring the effectiveness of data preparation steps. The development of improved deep learning techniques for parking lot occupancy detection has the potential to revolutionize real-time parking management. The project focused on various facets of deep learning, covering the fine-tuning of a ResNet18 model, custom model architecture, data preprocessing for a specialized dataset, and troubleshooting during model training. It delved into a project on parking lot occupancy detection, highlighting data-related processes and tailored model development. Lastly, it analyzed training metrics, gaining insights into model performance over epochs. This conversation encapsulates diverse aspects of deep learning, from architecture to training strategies and evaluation considerations for specific tasks.

# **References**

1. Debaditya Acharya, Weilin Yan, and Kourosh Khoshelham. Real-time image-based parking occupancy detection using deep learning. Research@ Locate, 4:33–40, 2018.
2. Ching-Chun Huang and Sheng-Jyh Wang. A hierarchical bayesian generation framework for vacant parking space detection. Circuits and Systems for Video Technology, IEEE Transactions on, 20:1770 – 1785, 01 2011. doi:10.1109/TCSVT.2010.2087510.
3. Hoang Tran Vu and Ching-Chun Huang. Parking space status inference upon a deep cnn and multi-task contrastive network with spatial transform. IEEE Transactions on Circuits and Systems for Video Technology, 29(4):1194–1208, 2019. doi:10.1109/TCSVT.2018.2826053.