ELEC 475 Lab 5 – Pet Nose Localization

ELEC 475 Prof. Michael Greenspan

Monday, December 5th, 2023

Mile Stosic (20233349)

Kieran Cosgrove (20226841)

Table of Contents

Table of Figures

1 Introduction

The purpose of this report is to outline the objectives and design challenges of the "Pet Nose Localization" initiative, a project under the ELEC 475 course being undertaken by Mile Stosic, and Kieran Cosgrove. This document is primarily intended for the Course Instructors and the Faculty Supervisor by highlighting the project's significance. It details the group's specific goals in addressing pet nose localization and the unique design problems they aim to solve, ensuring that the primary audience fully understands the project's scope and potential impact.

1.1 Project Scope

The objective of this project was to develop a deep learning model for the localization of pet noses in images. This task is significant in the field of computer vision, particularly in applications related to pet identification and tracking. The focus was to accurately predict the nose coordinates of pets in a given dataset.

1.2 Methodologies

The project employed a convolutional neural network (CNN) using the ResNet-18 architecture as the foundation. The approach involved fine-tuning a pre-trained ResNet-18 model, adapting it specifically for nose localization. The training and testing process utilized a custom dataset comprising images labeled with nose coordinates. The project incorporated techniques such as hyperparameter tuning and loss function optimization to enhance the model's performance.

2 Model

As aforementioned, the model is a customized convolutional neural network built upon the Resnet-18 architecture. The original Resnet-18 model is respected for its residual learning framework, which enables the training of deeper networks by using skip connections or shortcuts to jump over some layers [1]. The modifications we introduced are detailed as follows.

2.1 Resnet-18 Rational

The base of our model utilizes the pre-trained Resnet-18 provided by Torchvision's models module. This model comes with weights trained on the ImageNet dataset, offering a solid foundation for feature extraction [1]. By incorporating residual connections, which allow layers to learn modifications to the identity mapping rather than complete transformations, ResNet-18 facilitates more efficient training of deeper models [1]. This results in enhanced feature extraction capabilities, making it highly effective for tasks like object localization [1].

Our adaptation of ResNet-18 involved modifying the final fully connected layer to tailor it for the specific task of pet nose localization, moving from a general classification task to a more focused regression problem: Resnet-18 model was originally a 1000 class classification problem [1]. By leveraging the pre-trained weights on the ImageNet dataset, we ensured that our model benefited from a comprehensive and diverse range of features, providing a robust foundation for accurate localization.

2.2 Custom Regression Head

The classification head of the original Resnet-18 is replaced with a custom regression head tailored to our task. This regression head is a neural network module consisting of three main alterations. A dropout layer was configured with a probability of 0.5 to prevent overfitting by randomly zeroing some of the elements of the input tensor during training. Additionally, the fully connected layer that reduces the dimensionality from the number of features output by Resnet-18 to just two, corresponding to the x and y coordinates of

the pet's nose. Last, a batch normalization layer that normalizes the output of the fully connected layer, stabilizing the learning process and accelerating convergence.

2.3 Training and Optimization

During training, the input images undergo a series of transformations such as resizing, color jittering for data augmentation, random horizontal flipping, and normalization using ImageNet statistics. These transformations are crucial for the model to learn from a more general representation of the data, thus enhancing its ability to generalize.

To fine-tune the model, we employ the EuclideanDistanceLoss as our criterion. This loss function computes the Euclidean distance between the predicted and true coordinates, providing a direct measure of localization accuracy.

The model is set to fine-tuning mode, allowing the gradients to be computed for all layers, thus updating the pre-trained weights for our specific task. The use of the Adam optimizer leverages adaptive learning rates for different parameters.

In summary, the model architecture is a sophisticated blend of pre-trained feature extraction capabilities from Resnet-18 and a tailor-made regression head for precise localization. The training strategy, inclusive of advanced data preprocessing and fine-tuning techniques, ensures that the model is not only accurate in its predictions but also robust against overfitting and variance within the input data.

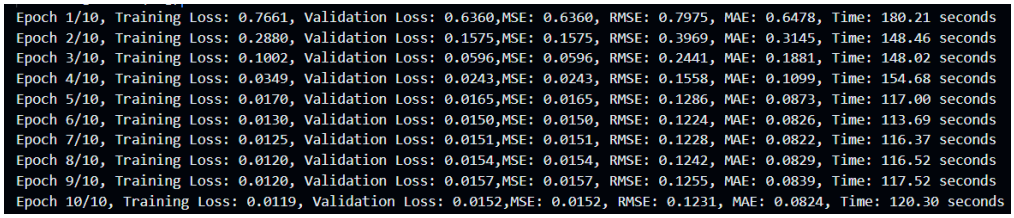

Figure 1: Training Terminal

2.3.1 Hyperparameters

For hyperparameter optimization, Optuna was employed to systematically explore and identify the best learning rate and optimizer for our model. The study encompassed 100 trials, thoroughly evaluating various combinations to pinpoint the most effective configuration.

The parameters chosen through Optuna's optimization process were pivotal in achieving the final model performance. The learning rate, determined to be optimal at 0.005588020278716033, was a crucial element in balancing the speed and stability of convergence during training. This value ensured efficient learning while avoiding potential issues such as overshooting the minimum loss.

In terms of the optimizer, the study concluded that Stochastic Gradient Descent (SGD) was the most suitable for this task. SGD's effectiveness in this context can be attributed to its ability to navigate the loss landscape effectively, avoiding local minima that could affect the learning.

The training process also incorporated an early stopping mechanism with a patience parameter set to 10. This approach allowed the training to stop if there was no improvement in the validation loss over the set number of consecutive epochs, preventing overfitting and ensuring computational efficiency.

These hyperparameters, carefully optimized and selected, were crucial in the training process towards an effective model capable of accurately localizing pet noses in various images.

```
[I 2023-12-05 14:11:13,288] Trial 0 finished with value: 0.43861699035001356 and parameters: {'lr': 5.795728515697265e-05, 'optimizer':
'Adam'}. Best is trial 0 with value: 0.43861699035001356.
[I 2023-12-05 14:20:34,730] Trial 1 finished with value: 0.18692612890587296 and parameters: {'lr': 0.05253032889648244, 'optimizer':
'RMSprop'}. Best is trial 1 with value: 0.18692612890587296.
[I 2023-12-05 14:30:18,767] Trial 2 finished with value: 0.051079805853754975 and parameters: {'lr': 0.06704908779006069, 'optimizer':
'SGD'}. Best is trial 2 with value: 0.051079805853754975.
[I 2023-12-05 14:40:25,381] Trial 3 finished with value: 0.10400546775307766 and parameters: {'lr': 0.005061855664205098, 'optimizer':
'SGD'}. Best is trial 2 with value: 0.051079805853754975.
[I 2023-12-05 14:50:41,182] Trial 4 finished with value: 0.11636058345090511 and parameters: {'lr': 0.008582502078181843, 'optimizer':
'RMSprop'}. Best is trial 2 with value: 0.051079805853754975.
[I 2023-12-05 14:51:40,597] Trial 5 pruned.
[I 2023-12-05 14:52:44,739] Trial 6 pruned.
```

Figure 2: Optuna Learning Parameters

2.3.1.1 Hardware

We used a local laptop GPU. Utilizing a NVIDIA GeForce GTX 1650 GPU for pet nose localization offers significant advantages, particularly in uninterrupted computational capability and data privacy. The Turing architecture of the GPU efficiently handles the extensive calculations required for deep learning tasks without incurring the latency and potential security concerns associated with cloud computing. Moreover, the local GPU setup bypasses potential cloud service costs and bandwidth limitations, facilitating a more cost-effective and seamless research and development process. With the ability to configure and control the system to meet project-specific needs, a local GPU provides an environment that closely mirrors real-world deployment scenarios, allowing for accurate performance assessment and optimization.

2.3.2 Training Time

The training of the pet nose localization model displayed efficiency, with an average of around 148 seconds per epoch and a total training duration of approximately 25 minutes for 10 epochs. This indicates a well-optimized training process suitable for rapid development cycles. These results are displayed above in Figure 2.

2.3.3 Loss Curve

The training process was monitored using a loss curve, which depicted the model's learning progress over epochs. The Euclidean Distance Loss function was implemented, offering a more intuitive measure of localization error compared to traditional MSE loss.

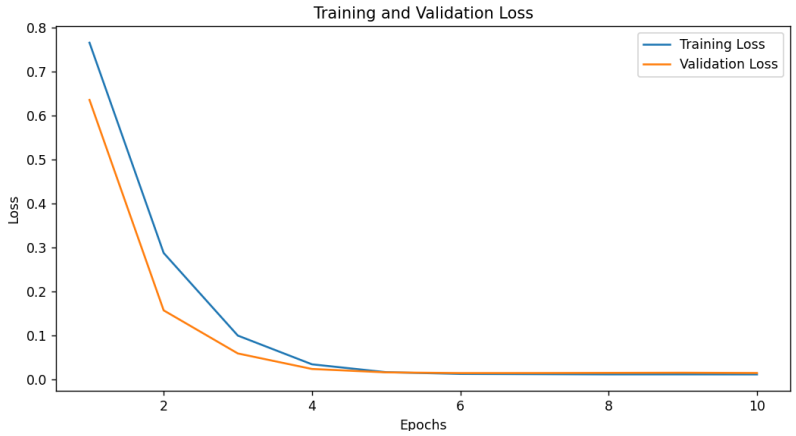

Figure 3: Training and Validation Loss Curve

The loss curve displayed a downward trend, indicating that the model was learning effectively over time. The training loss started at 0.7661 and gradually decreased to 0.0119, whereas the validation loss began at 0.6360 and stabilized around 0.0152.

2.4 Testing

The testing phase involved evaluating the model's performance on unseen data. The results were assessed both qualitatively, by visual inspection of the predicted nose coordinates, and quantitatively, using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

The testing script loaded the pre-trained model and ran inference on the test dataset, comprising images with known nose coordinates. The predictions were compared against these true values to calculate the accuracy statistics. The process involved normalizing the coordinates to a range [0, 1], relative to the image dimensions, to provide a standard scale for the loss calculations during training. For testing, the predicted coordinates were denormalized back to the original image scale using the image dimensions to calculate the evaluation metrics accurately.

3 Results

This section showcases the quantitative and qualitative results from the execution of this lab, and it illustrates the success of our project.

3.1 Qualitative Results

The visual examination of test images shows a correlation between the predicted and true labels, with most predictions falling within an acceptable range of the actual nose positions, however some did not fall within an acceptable range.

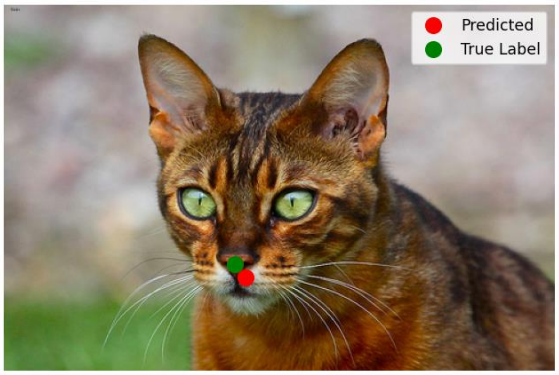

Figure 4: First Nose Localization Example

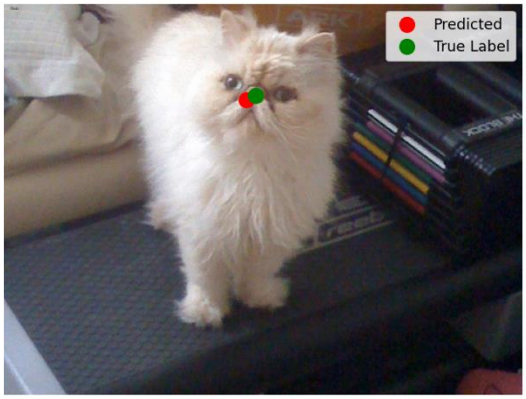

Figure 5: Second Nose Localization Example

Figure 6: Third Nose Localization Example

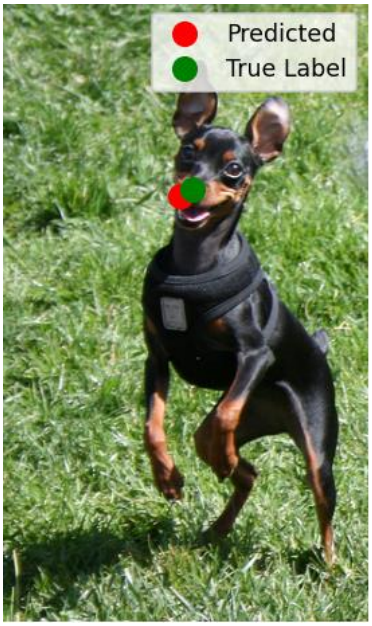

Figure 7: Fourth Nose Localization Example

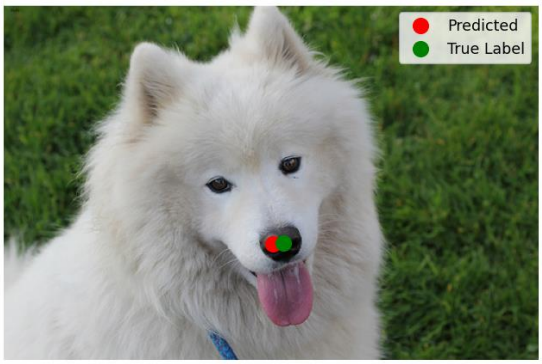

Figure 8: Fifth Nose Localization Example

Figure 9: Sixth Nose Localization Example

3.2 Accuracy Statistics

The performance of the nose localization model was measured using standard metrics, and the results underscore the model's high degree of accuracy. The MSE achieved was 0.0152, which reflects the average of the squares of the errors and is a measure of the quality of the estimator; it is always non-negative, and values closer to zero are better. The RMSE, calculated at 0.1231, provides insight into the magnitude of the error made by the model in predicting the nose coordinates. The MAE, observed at 0.0824, indicates the average absolute difference between the estimated values and the actual values, providing a linear score that represents the average magnitude of the errors in a set of predictions, without considering their direction.

The low MSE value points towards minimal variance in the model's predictions, suggesting a high level of precision. Similarly, the RMSE, being the square root of MSE, implies a low error margin in the model's predictions with respect to the scale of the target values. This is particularly important as it provides a more interpretable measure of prediction accuracy. Lastly, the MAE offers an average error

magnitude per dataset sample, and its low value is indicative of the model's ability to perform highly accurate predictions with a small average deviation from the true labels.

To further measure the performance of the model's accuracy, a set of test images with annotated predictions was reviewed and seen above in the figures above. The proximity of the predicted (red dot) to the true labels (green dot) was consistently close across the majority of the test cases, providing qualitative evidence to support the quantitative metrics. This close alignment is a testament to the model's robustness and its sophisticated understanding of the nuanced task of nose localization.

In summary, the accuracy statistics, joined with qualitative visual validations, confirm the efficacy of the model, making it a strong candidate for practical applications requiring nose localization in pets.

4 Discussion & Conclusion

We anticipated high accuracy from the fine-tuned ResNet-18 model based on its strong image recognition capabilities. While the model generally performed well, as shown by low error metrics, there were occasional inaccuracies in localizing pet noses.

The CNN model exhibited promising results in pet nose localization, yet it faced challenges with certain images. These challenges can be attributed to factors such as diverse pet poses, occlusions, variable lighting conditions, and complex backgrounds. The observations underscore the need for a robust and diverse training dataset, as well as the potential benefits of advanced image preprocessing and model architectures.

To overcome challenges in pet nose localization, we fine-tuned the ResNet-18 model, employed a custom Euclidean distance loss function, optimized hyperparameters via Optuna, and introduced data transformations like resizing and color jitter. These strategies enhanced the model's learning efficiency and robustness, enabling better adaptation to diverse pet images and improving generalization capabilities. Continuous testing and validation were key in refining the model's accuracy.

4.1 Performance Observations & Recommendations

Future recommendations include expanding the dataset to cover a wider range of scenarios, employing more sophisticated neural network models for feature extraction, and exploring post-processing techniques to refine predictions. With these enhancements, the model's accuracy and generalization across various images are expected to improve, leading to more consistent and reliable localization performance.