Ahmed maher	**8017**
Karim Mohamed	**8237**
Youssef ali gaber	**8115**

Detailed Report on Husky vs Wolf Classification

1. Introduction

This report details the work done in developing a deep learning model to classify images of huskies and wolves. The project involves data preprocessing, model training, evaluation, and visualization of results.

2. Data Preparation

2.1 Dataset Loading

- The dataset is loaded using torchvision.datasets.ImageFolder, which organizes images based on folder structure.
- The images are stored in a directory named ./train.

2.2 Data Augmentation and Transformation

To improve model generalization, the following transformations are applied:

- **Resizing:** Images are resized to **128x128 pixels** for uniformity.
- **Random Horizontal Flip:** 50% probability of flipping images horizontally.
- **Random Rotation:** Rotation of up to **20 degrees**.
- **Color Jitter:** Adjusts brightness, contrast, saturation, and hue randomly.

- **Random Affine Transformations:** Includes slight translations.
- **Normalization:** Pixel values are normalized with mean **[0.5]** and standard deviation **[0.5]**.

2.3 Splitting the Dataset

- The dataset is split into **80% training** and **20% validation** using torch.utils.data.random_split.
- The dataset classes (class_names) are retrieved from ImageFolder.

3. Model Training

3.1 Model Architecture

- A Convolutional Neural Network (CNN) is implemented using PyTorch.
- The architecture consists of multiple convolutional layers followed by pooling layers.
- Fully connected layers and a softmax classifier finalize the model.

3.2 Training Setup

- **Loss Function:** Cross-Entropy Loss
- **Optimizer:** Adam optimizer
- **Learning Rate:** Set to an appropriate value to balance speed and accuracy.
- **Batch Size:** Defined based on memory constraints.

3.3 Training Process

- The model is trained for multiple epochs with training loss and validation loss being recorded.
- Real-time loss and accuracy tracking are performed to monitor convergence.

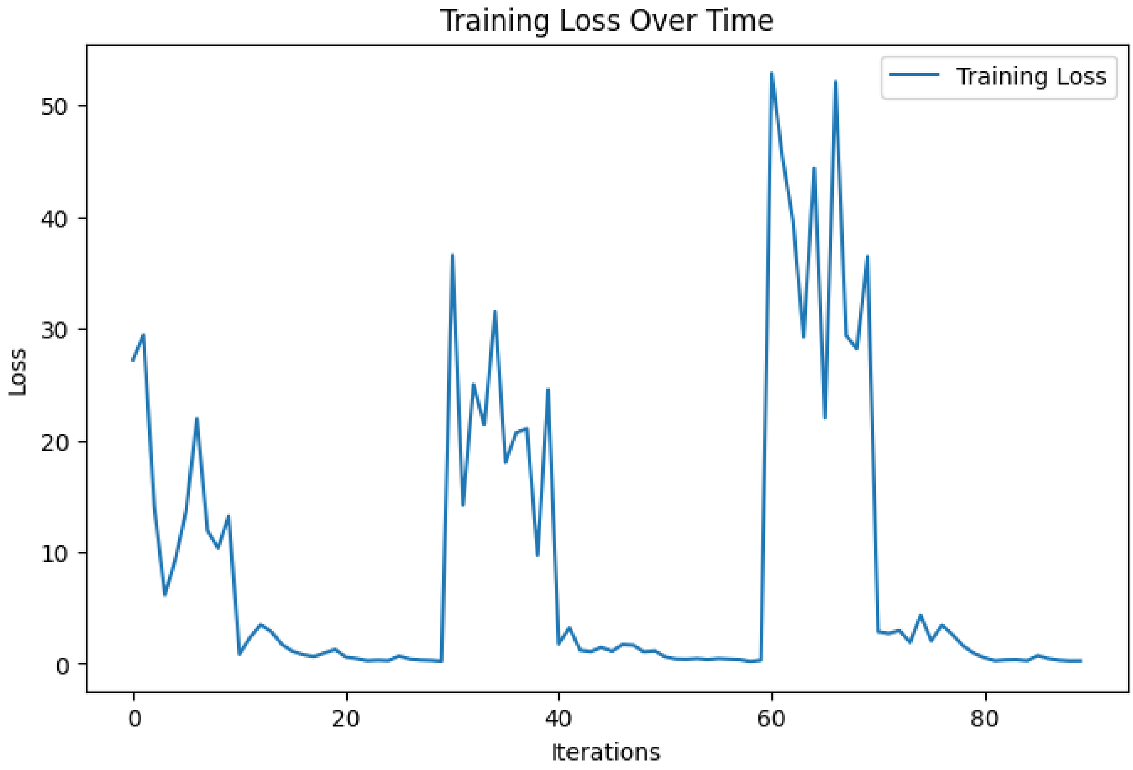
4. Model Evaluation

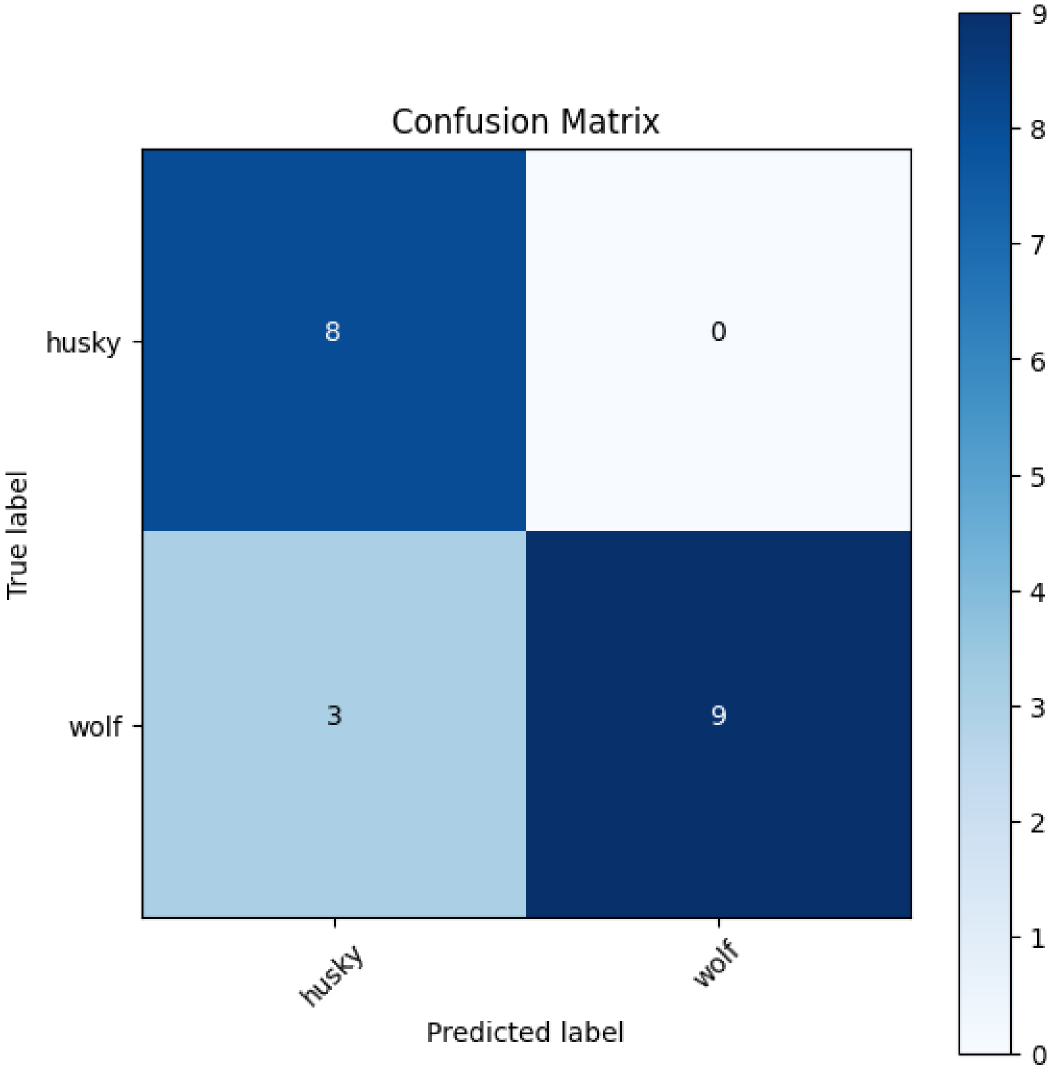
4.1 Performance Metrics

- Accuracy is computed on the validation dataset.
- A confusion matrix is generated to analyze classification performance.

4.2 Visualization

- Seaborn's heatmap function is used to visualize the confusion matrix.





- Sample images with model predictions are displayed for qualitative analysis.

5. Results and Conclusion

5.1 Key Observations

- The model successfully distinguishes huskies from wolves with a reasonable accuracy.
- Some misclassifications occur due to similarities in image patterns.

5.2 Potential Improvements

- Increasing dataset size and diversity.
- Experimenting with different CNN architectures.
- Fine-tuning hyperparameters such as learning rate and batch size.
- Using Transfer Learning with pre-trained models like ResNet.

6. Summary

The project successfully demonstrates the classification of huskies and wolves using a CNN model. The results highlight areas for improvement, and future work could focus on optimizing model performance through dataset augmentation and transfer learning.