IBA SMCS
School of Mathematics and Computer Science

CSE669 Deep Learning course

Institute of Business Administration, Karachi

School of Mathematics and Computer Science Department

Assignment 03: Dog vs Cat vs Bird

Author:
Mohammad Kamil (ID: 29464)

Date: December 25, 2024

Contents

1 Introduction

Image classification is A fundamental application of computer vision, and it is essential to many fields, such as healthcare, retail, and autonomous systems. It automates processes that formerly required human involvement by enabling machines to examine and understand visual input. Convolutional neural networks (CNNs) is a type of deep learning approach which have revolutionized the discipline in recent years by obtaining state-of-the-art performance on image classification tasks.
This assignment aims to create an image classification model that can accurately recognize images of dogs, cats, and birds. These categories are a multi-class classification problem that presents difficulties such as intra-class variations, visual similarity between classes and complicated data preprocessing.
The implementation was carried out using conventional CNN architectures and Transfer Learning with pre-trained models like ResNet-18 and VGG-16. These architectures were implemented to increase accuracy while preserving computing power. More sophisticated optimization methods, such as gradient clipping, learning rate scheduling, and hyperparameter tuning, were used to improve the model's performance.
Furthermore, predictions from several models were combined using ensemble learning, which successfully reduced the biases of each model but didn't achieve good accuracy. However, ResNet-18 gives better prediction accuracy than ensemble methods which is 88%.
This report presents a detailed overview of the methodology, covering the construction of baseline models, optimization strategies, transfer learning tactics, and data set preprocessing and exploration. Performance metrics and insights are discussed thoroughly in this report.

2 Problem

Image classification has become a crucial application of machine learning and computer vision in the age of rapidly developing artificial intelligence. Several industries including healthcare, agriculture, retail, and security, depend on the ability to identify objects in images accurately. Model performance is still impacted by issues including intraclass variability, interclass similarity, and the complexity of data preprocessing, even with notable advances in image classification.
This assignment aims to create and use an image classification model that could tell the difference between three visually similar groups: birds, dogs, and cats. Since every category has different visual patterns, the model must be able to learn and generalize across various features.
Image classification systems have several uses in both business and academia. Putting images into meaningful categories accurately can improve decision-making, accelerate processes, and reduce human error. By tackling the issues mentioned in this assignment,

I developed a model by creating a reliable, scalable, and compelling image classification model.
The advancement of image-based decision systems in fields like wildlife monitoring, surveillance systems, and medical imaging diagnostics where object recognition is crucial, is based on this challenge.

3 Dataset Preprocessing and Exploration

In this section, I discussed the dataset downloaded from the Kaggle competition and its further exploration.

3.1 Data Overview

A training set and a test set are the two main subsets of the dataset I downloaded from our class's Kaggle competition. There are 40000 labeled images in the training set and 20000 unlabeled images in the test set as show in *Table 1*. Dogs, cats, and birds are the three different classes these images are divided into training and testing. The format of very image is saved as a PNG file and is supplied in a standard 32x32 pixel format. While the unlabeled test images are utilized for model evaluation and performance assessment,the annotated training images aid in supervised learning.

Table 1: Dataset Overview

Subset	**Number of Images**	**Labels Available**	**Classes**
Training Set	40000	Yes	Dogs, Cats, Birds
Test Set	20000	No	N/A
Total	**60000**		

3.2 Data Preprocessing

Many necessary steps were taken during the dataset's preprocessing to ensure that the images were in the good format for training and evaluation. A uniform 64x64 pixel resolution was first applied to every image. By standardizing the input dimensions throughout the dataset, this resizing phase ensured that the neural network architecture would work. Pixel values were standardized after scaling using a mean and standard deviation of 0.5. This normalization phase placed all pixel values into a single range, which enhanced model stability during training and improved the performance of different models as shown in *figure 1* where have shown all the necessary steps that needed for this assignment.
Data augmentation techniques were used just on the training set to increase the diversity of the training data and boost the model's ability for generalization. Random

rotations up to 10 degrees, which added tiny rotational variations and random horizontal flips that mirrored the images along the vertical axis were typical of these augmentations. Color jittering was added which made minor alterations to the contrast and brightness of images. The combined goal of these preprocessing stages was to produce a more resilient and flexible model that could deal with changes in actual data.

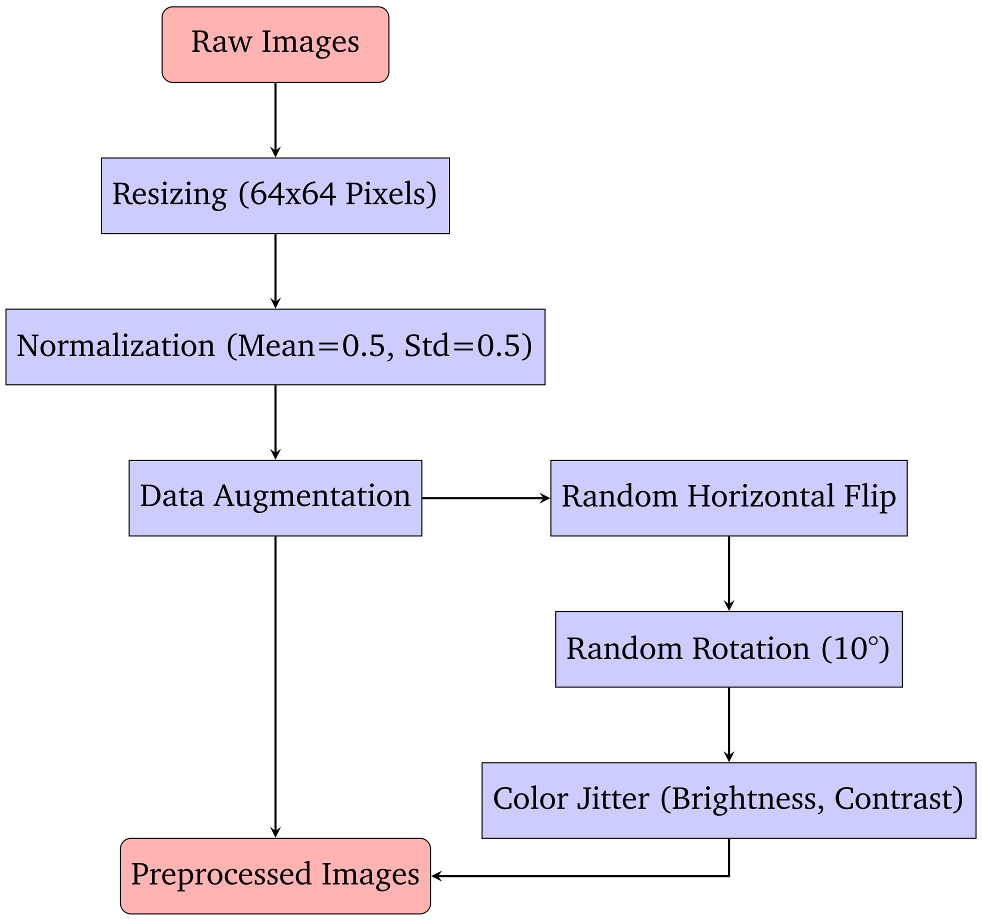

Figure 1: Data Preprocessing steps

3.2.1 Balanced dataset

The data set used in this assignment has a balanced distribution with an equal amount of images from each of the three classes, cats, dogs, and birds. To prevent any one category from controlling the dataset 4000 images are used to represent each class. During training and evaluation the likelihood of model bias towards any certain class is reduced because to this balanced distribution. It makes predictions more accurate and impartial for all classes by enabling the deep learning models to learn characteristics equally from all categories. The three categories are uniformly represented, as *figure 2* shows providing that the model's performance metrics fairly show its capacity to generalize across classes and not be influenced by class imbalance.

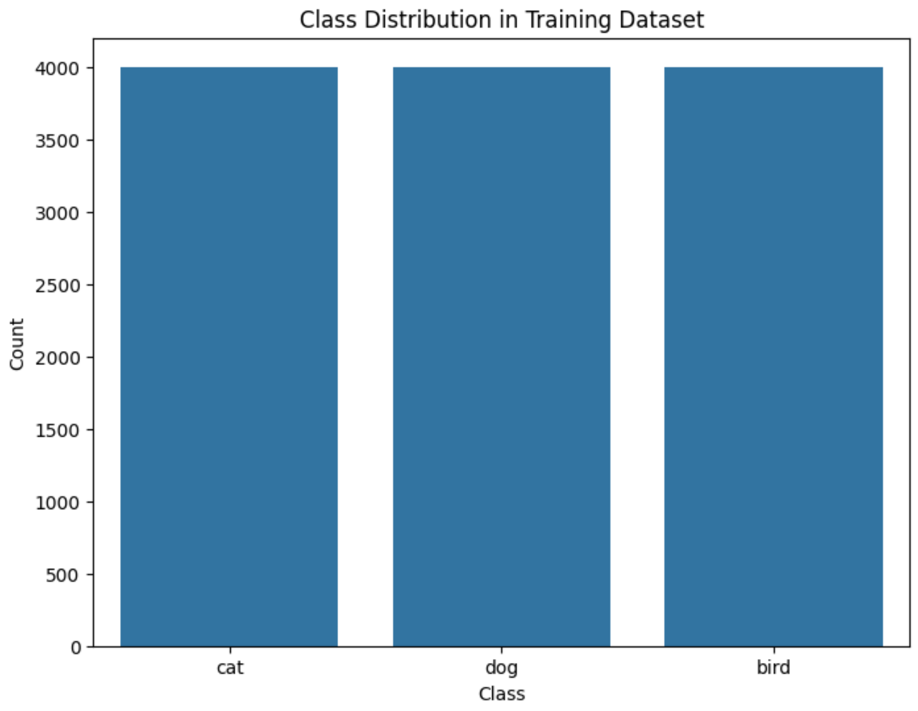

Figure 2: Class Distribution of the Dataset

3.2.2 Dataset Split and Distribution

Three subsets of the dataset were carefully separated to ensure efficient model evaluation, validation and training. The model primarily uses the training set which consists of 9600 images to identify patterns and characteristics. During training the validation set which consists of 2400 images utilized to adjust model parameters and avoid overfitting. The 3000 images test set is used to assess how well the model performs on unknown data as shown in *figure 3* . The distribution of the three classes cats, dogs, and birds is kept balanced within each subgroup.

3.2.3 Image Dimensions Analysis

The image dimensions of the dataset were checked for consistency following preprocessing to ensure compatibility with the deep learning model architecture and image dimensions are very important when we train our models. A random sample of 500 images from the training data was selected and analyzed to verify that each image had been uniformly resized to 32x32 pixels in PNG format. This standardization helps avoid

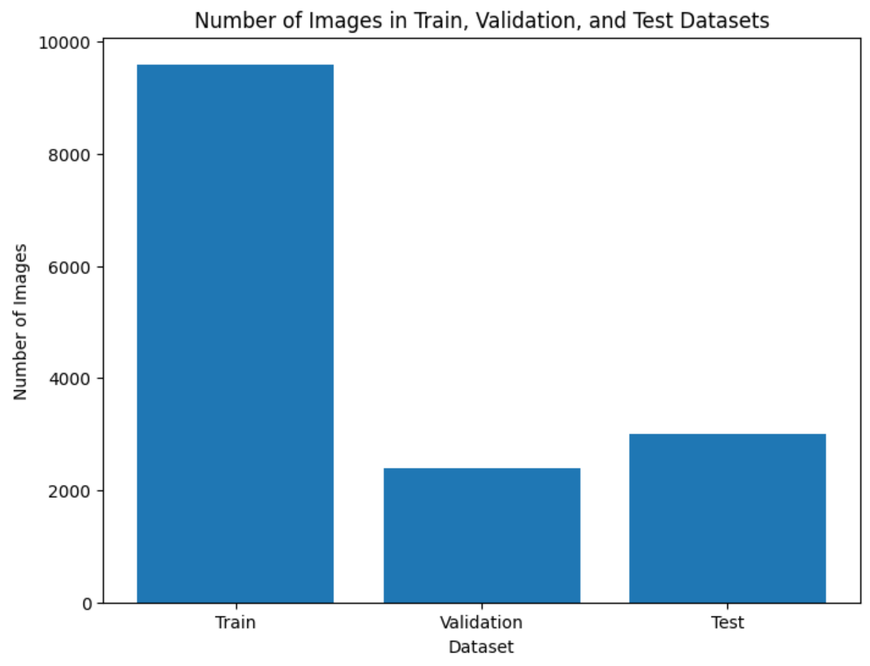

Figure 3: Dataset Split into Training, Validation and Testing

errors caused by inconsistent input dimensions also it reduces computational computational and ensures that each image aligns with the model's input requirements and can easily train on these dimensions image data. It ensure uniformity across the dataset and enabling smoother training and better performance on these models. The visualization of image dimensions is shown in *figure 4* where all selected images consistently maintain a 32-pixel width and 32-pixel height.

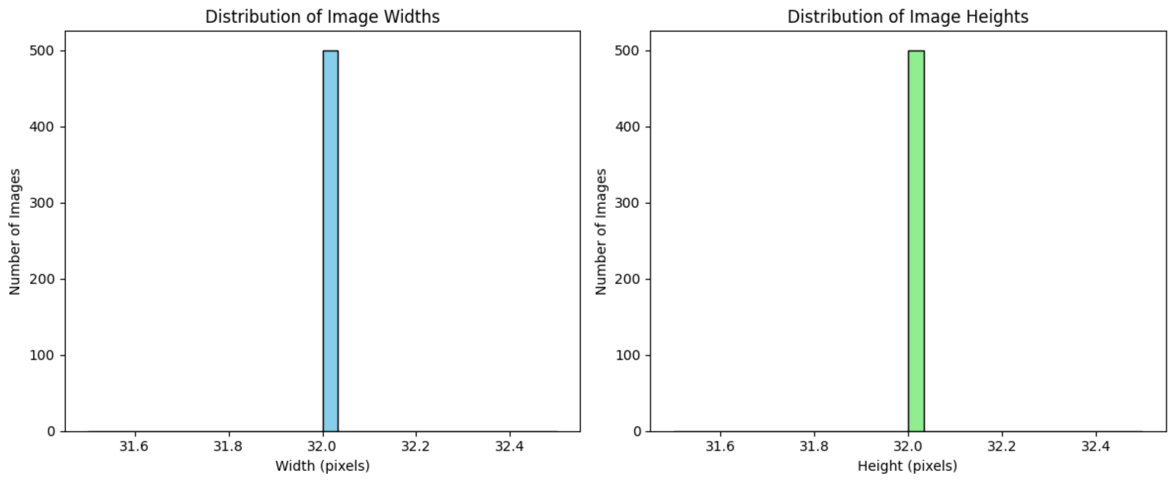

Figure 4: Distribution of Images Widths and Heights

3.2.4 Visualization of Sample Images from Each Class

To confirm the quality of images across several categories and have a better understanding of the visual variation in the dataset a grid of sample images was created for each class as shown in *figure 5*. A randomly chosen subset of images from the Cats, Dogs and Birds classes is shown in this structured grid where each row corresponds to a certain class and each column features several examples. The visualization provides an intuitive picture of the general makeup of the dataset by enabling a thorough evaluation of image quality, class representation and consistency between samples.
It also acts as a check to make sure that preprocessing steps like resizing and normalization were applied to every image correctly. Before moving on to the training stage, possible anomalies, including warped images, incorrect resizing, or inconsistent data labeling, can be identified and corrected using this visualization.
Because of preprocessing and resizing effects, the images in the grid are not entirely clear, but they are still distinct enough to identify important visual characteristics of each class. The grid helps identify any possible outliers or incorrectly classified images which may otherwise add noise to the training process and impair model performance. Intra-class variability such as variations in image lighting, angles, or object placement—may affect how effectively the deep learning model generalizes to new data. This can be seen through the visualization.
As shown in *figure 5* where each row corresponds to one class while the columns display multiple representative images. This layout highlights the distinct visual characteristics of each category and ensures balanced representation across all classes.

4 Baseline Model Development

In this section, I have discussed model architecture, loss functions, learning rates, and baseline model results and implementation.

4.1 Model Architecture

As a baseline model I used Convolutional Neural Networks (CNNs) in the development of image classification model in order to effectively process picture data and extract significant features for classification. The first two convolutional blocks in the design are made up of a Conv2D layer, a MaxPooling layer, and a ReLU activation function. 32 3x3 filters are applied by the first convolutional layer, and 64 filters are applied by the second. In order to ensure computing efficiency, these layers are in charge of capturing spatial elements like edges, forms, and textures while lowering the dimensions of space through pooling operations.
The feature maps are flattened and run through fully connected layers after the convolutional layers. In order to avoid overfitting, the first fully linked layer has a Dropout layer (0.5) and 256 neurons with a ReLU activation function. Predictions for the three target

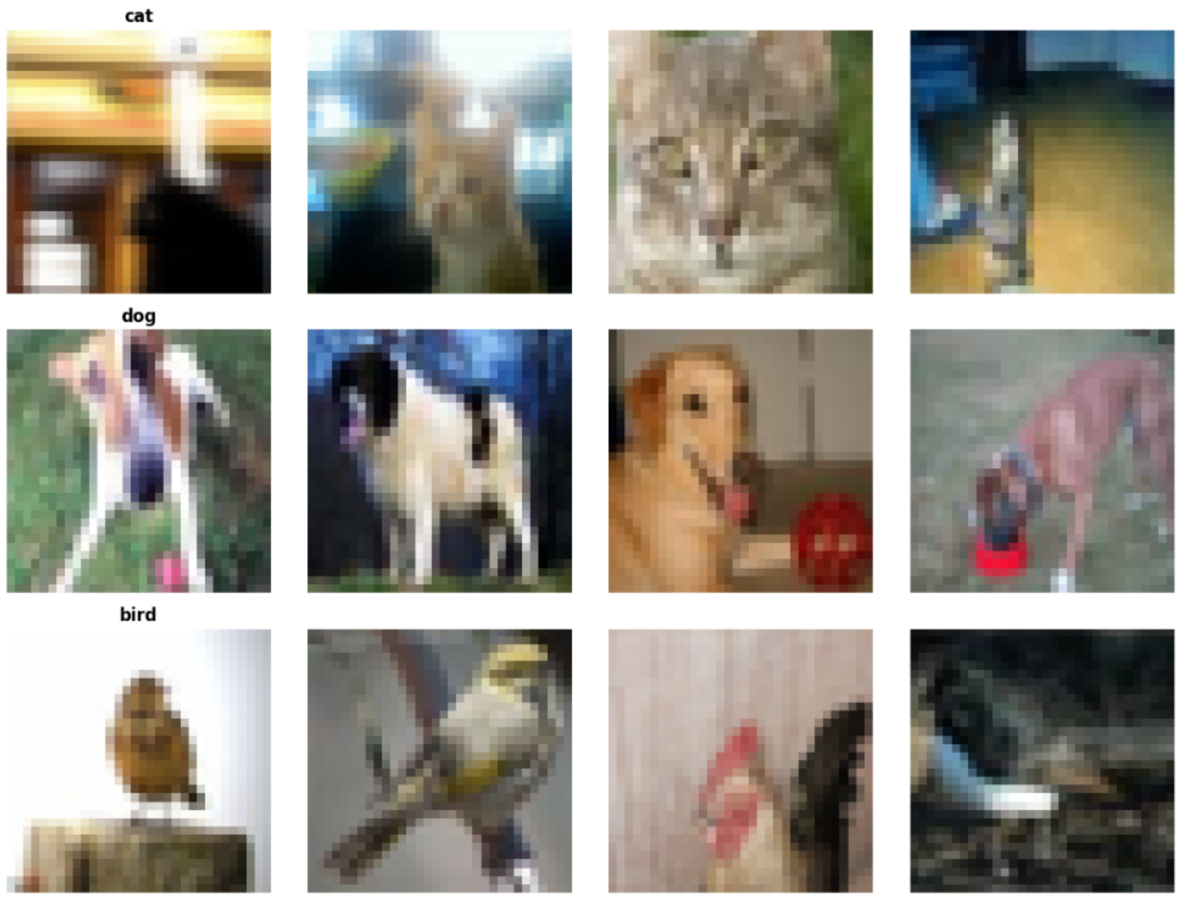

Figure 5: Visualization of Sample Images from Each Class

classes like cats, dogs, and birds that are produced by the last layer. for developing this architecture I used PyTorch which aims to strike a balance between accuracy and computing efficiency. *Figure 6* provides a visual representation of the model's structure, displaying the data's sequential progression through convolutional, activation, pooling, dropout, and fully connected layers.

4.2 Loss function and learning rate

The CNN model was trained using the CrossEntropyLoss as the loss function as it calculates a difference between the actual class labels and predicted class probabilities. This loss function works well for multi-class classification tasks and the model learns to make better predictions by reducing this loss.
With a learning rate of 0.001, the Adam optimizer was used for optimization.It makes it possible to handle sparse gradients well and adjust learning rates for every parameter. Throughout the training process the model is able to learn effectively and consistently

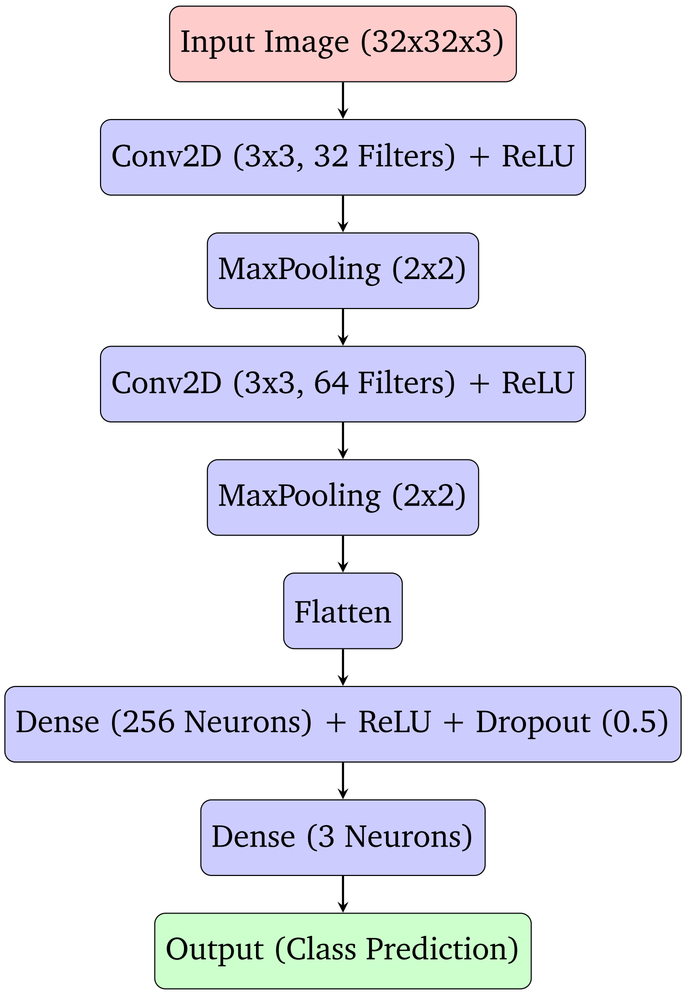
Input Image (32x32x3)
Conv2D (3x3, 32 Filters) + ReLU
MaxPooling (2x2)
Conv2D (3x3, 64 Filters) + ReLU
MaxPooling (2x2)
Flatten
Dense (256 Neurons) + ReLU + Dropout (0.5)
Dense (3 Neurons)
Output (Class Prediction)

Figure 6: Baseline CNN Architecture

because to the combination of the CrossEntropyLoss and Adam optimizers.

4.3 Baseline model Results

The baseline CNN model evaluated with varying combinations of learning rates, dropout rates, and number of epochs to analyze their impact on validation accuracy. The learning rate determines the step size during weight updates while dropout prevents overfitting by randomly deactivating neurons during training. The number of epochs represents the number of complete iterations over the dataset during training.
A fixed dropout rate of 0.5 was used to test two distinct learning rateslike 0.001 and 0.0001 so throughout two training durations 10 and 20 epochs. So as per my results the learning rate of 0.001 continuously performed better than 0.0001 over both training periods, exhibiting improved convergence and increased validation accuracy. I have

shown the detailed results in *Table 2*.

Table 2: Baseline CNN Model Results with Different Hyperparameters

Learning Rate	Dropout Rate	Epochs	Validation Accuracy
0.001	0.5	10	0.67
		20	0.72
0.0001	0.3	10	0.64
		20	0.69

So from all these results, the baseline model performed well with a learning rate of 0.001, dropout of 0.5, and 20 training epochs and achieved 67% accuracy.

5 Optimization Techniques

As I haven't get good results in baseline model with hyperparametres so before implementing transfer learning models i implement different advance optimatization techniques to get good results with transfer learning models.

1. **Learning Rate Scheduler (StepLR)**

   The learning rate was lowered every seven epochs using the learning rate scheduler. This modification ensures that the model may make significant changes during the first training period, with small refinements being performed in later stages. In order to achieve stable converging and increase validation accuracy, this technique was implemented and transfer learning models shows good improvement.

2. **Weight Decay (L2 Regularization)**

   Weight decay was used as an L2 regularization factor in the loss function to avoid overfitting. By penalized high weight values, this approach encourages the model to learn more simple and broadly applicable patterns instead of learning the training data's noise. It had a direct impact on enhancing the model's performance on hidden validation data.

3. **Dropout Layers**

   During training, dropout layers were added to the fully linked layers to randomly deactivate neurons. This method improves generality by preventing the model from depending too much on particular neurons. Particularly in the model's deeper layers, it significantly reduced the likelihood of overfitting.

4. **Gradient Clipping**

To restrict the maximum value of gradients during backpropagation, gradient clipping has been used. Exploding gradients, which can interfere with training, particularly in deeper networks, are avoided with this method. It helped continually improve performance by ensuring more stable weight updates and smooth training.

6 Transfer Learning

As we have image classification problem of identifying between cats, dogs and birds so i used transfer learning learning models in order to get good results as compared to baseline models. Using pre-trained models on huge data sets (like ImageNet) this method refines them for the particular classification goal. Because basic visual characteristics like edges, textures, and patterns are already captured by the first layers of pre-trained models, transfer learning significantly cuts down on training time and increases accuracy.

6.1 Pre-trained Models

I have chooses two pre-trained architectures like ResNet-18 and VGG-16 for this image classification work because of their shown ability to handle complex image datasets. By utilizing learned representations from millions of images, these models which have already been trained on the ImageNet dataset offer a solid basis and remove the need for intensive training from the beginning.

1. In order to preserve gradient flow during backpropagation, I choose ResNet-18 due to its capacity to manage deeper structures through residual connections. Both low-level and high-level features can be captured by it without adding excessive processing load owing to its effective architecture.

2. The sequential and consistent architecture of VGG-16 made fine-tuning easier and allowed it to adapt effectively to picture classification challenges. Because of its depth, it can acquire complex spatial features, which makes it a dependable option for correctly differentiating between dogs, cats, and birds.

The fully connected layers of both models have been modified and optimized for the given task. In contrast, the initial convolutional layers of both models were frozen to maintain pre-trained feature representations when they were merged into the pipeline. These structures provided experts a solid starting point for testing while generating encouraging outcomes regarding accuracy and dataset generalization.

7 Optimization of Transfer Learning

In this section I have discussed different optimization techniques and discussed fine-tuning in detail.

7.1 Fine-Tuning

A carefully developed methodology was used to optimize transfer learning models, namely ResNet-18 and VGG-16, in order to ensure robust training, avoid overfitting, and optimize model performance. The expected increases in classification accuracy were achieved by combining regularization methods, training strategies, learning rate scheduling and optimization algorithms.

To start the optimization process, a CrossEntropyLoss function was combined with the Adam optimizer, which is known for its adjustable learning rates to tackle the multi-class classification task. To overcome the limitations of weight decay in traditional Adam optimization, the AdamW optimizer was also used. The pre-trained models were fine-tuned purely on the completely connected layers with the starting layers frozen. Learning rate schedulers proved crucial in helping to further regulate the learning process. To enable more accurate weight changes in future epochs, a StepLR scheduler was first used to lower the learning rate after every seven epochs. A Cosine Annealing Scheduler was added in later tests which helped the models converge more smoothly towards optimal solutions by gradually lowering the learning rate in line with a cosine curve.

Regularization strategies were also used to ensure improved generalization. To lower the chance of overfitting, dropout layers were included to the fully linked layers. These layers randomly deactivated a portion of the neurons during training. The model was further encouraged to learn simpler patterns by penalizing big weight values using Weight Decay.

In order to prevent rising gradients particularly in deeper layers of the models gradient I used clipping to cap the gradients within a certain range and preserve training stability. MixUp augmentation was added to the training loop in after optimization phases to enhance generalization by combining training samples and their corresponding labels, which reduces the model's dependence on particular data points. By gradually using these strategies, the training loop was gradually improved. While learning general patterns through frozen layers was the main focus of early training epochs, later epochs focused on fine-tuning the entire network with reduced learning rates. The total model performance and validation accuracy gradually improved as a result of these combined methods.

The results of all these models and techniques is mentioned in next section that how transfer learning models perform on image classification problem with different parameters and techniques.

8 Results, Metrics and Insights

In this section, I thoroughly discussed the results and insights about different models.

8.1 Model Evaluation and Results

This section shows the performance evaluation of every model used in the assignment from the Baseline CNN Model to more complex architectures like the final Ensemble Model and Transfer Learning Models (ResNet-18, VGG-16).The primary evaluation metric applied to all models is Accuracy which gives an accurate picture of how well each model classifies images into the three groups of cats, dogs, and birds. From all these models I got highest accuracy with ResNet-18 that is 88% as shown in *table 3*.
The Adam optimizer, Learning Rate Scheduling, Dropout layers, Weight Decay, Gradient Clipping, and Batch Normalization are among the optimization techniques used to train and improve each model. Interestingly, changing epochs sizes significantly affected both final accuracy and training consistency for all models.

Table 3: Comparison of Models Performance

Model	Learning Rate	Epochs	Dropout Rate	Accuracy
Baseline CNN	0.001	10	0.5	0.67
Baseline CNN	0.001	20	0.5	0.72
Baseline CNN	0.0001	10	0.3	0.64
Baseline CNN	0.0001	20	0.3	0.69
ResNet-18	**0.001**	**10**	**0.5**	**0.88**
ResNet-18	0.0001	10	0.3	0.77
VGG-16	0.001	10	0.3	0.67
VGG-16	0.001	20	0.5	0.0.79
Ensemble Model	0.001	10	0.5	0.85

So, from all models, the results of ResNet-18 showed the highest performance, and after that, the ensemble model showed good performance.

8.2 Insights from Results

As I have implemented different models with different techniques and parameters and the analysis of the implemented models, from the Baseline CNN to ResNet-18, VGG-16, and the Ensemble Model reveals several key insights about model performance and behavior during training.

The effect of learning rate on accuracy and convergence is one main finding. A moderate learning rate ensures improved convergence and avoids unduly slowly training, as shown by the enduring greater performance of 0.001 over 0.0001. Accuracy was typically increased by increasing the number of epochs for all models, with VGG-16 requiring more epochs to utilize its deeper design properly. ResNet-18 showed quicker convergence due to its residual connections, which helped to reduce gradient-related issues during backpropagation.
Another important factor in the generalization of the model was the dropout rate. Compared to 0.3, a dropout rate of 0.5 improved regularization and decreased overfitting, particularly during longer training runs. This highlights how crucial it is to choose the right dropout values to avoid overfitting and maintain the model's ability to learn.
Because of its effective architecture, ResNet-18 regularly exceeded VGG-16 in terms of accuracy but VGG-16 required more training epochs and careful tuning to produce equivalent results. By combining predictions from ResNet-18 and VGG-16, the Ensemble Model provided balanced performance, utilizing each model's benefits to increase accuracy and adaptability.
Lastly, methods like Gradient Clipping, Weight Decay, and Dropout Regularization were crucial for preserving stable training and avoiding gradient-related problems and overfitting. These results highlight how important it is to adjust hyperparameters, use appropriate architectures and use ensemble techniques in order to provide accurate and reliable picture classification results.

9 GitHub Repository

I have added an implementation file with a properly formatted GitHub repository where i added all the required details like the dataset overview and implementation file, including evaluation, model development, optimization techniques, and dataset preprocessing. This repository provides a thorough resource for verifying the findings and expanding on the experiments conducted out in this assignment.

GitHub Repository Link