CSE4288 Introduction to Machine Learning
Team Project Fall 2024
Final Project Report

Team Members:

1- Eren Duyuk 150120509
2- Ufuk Acar 150121071
3- Yusuf Demir 150120032
4- Emir Uyar 150120007
5- Muhammed Hayta 150121068

Abstract

This project focuses on creating a machine learning model that classifies children's hand-drawn images into ten categories, such as animals, plants, and everyday objects. Using a convolutional neural network (CNN), we tackle the unique challenges of children's drawings, which are often abstract and vary widely in style. To enhance the model's performance, we implemented data preprocessing and augmentation techniques, ensuring the model's adaptability and accuracy. Multiple algorithms were also tested for comparison, demonstrating the effectiveness of our CNN-based approach. This project highlights the potential of machine learning to bring children's creativity to life in interactive and meaningful ways.

Introduction

Children's drawings are a great expression of creativity, filled with unique shapes, imaginative details, and a wide variety of styles. Building a machine learning model to classify these drawings is not just a technical challenge but also an opportunity to bridge the gap between creativity and technology. This project aims to develop an image classification model capable of recognizing drawings of objects like birds, cars, and trees, using state-of-the-art techniques in deep learning.

Our dataset features a mix of authentic children's artwork and hand-drawn images sourced online, providing a diverse and balanced set of examples across ten categories. To prepare the data, we applied detailed preprocessing steps, such as bounding box labeling and data augmentation, ensuring the model can handle the inherent variability of these drawings. Leveraging a convolutional neural network (CNN), we crafted a model designed to excel in this challenging but rewarding task.

The significance of this project lies in its application to the entertainment sector. Specifically, this model is designed for an augmented reality (AR) game where children's drawings can be brought to life. The game will recognize the drawings and place 3D models on top of them, creating an immersive and interactive experience that encourages creativity and engagement. Additionally, this project has broader implications as a tool for education and child development. By providing insights into children's cognitive and artistic growth, it can also aid in automated feedback and categorization, supporting both learning and creative expression.

Methodology

This project employed a comprehensive methodology encompassing data preprocessing, model development, and evaluation to create a robust system for classifying children's drawings. Each stage was carefully designed to address the unique challenges posed by the dataset, such as variability in drawing styles and the abstract nature of the images. Baseline models were utilized alongside a Convolutional Neural Network (CNN) to ensure a thorough understanding of performance and to benchmark effectiveness.

Data Preprocessing

To prepare the dataset for effective model training and evaluation, the following steps were undertaken:

1. **Data Cleaning**:
    a. Images were labeled using Label Studio. Bounding boxes were drawn around the objects of interest, and corresponding class labels were assigned to each object.
    b. Mislabelled or unclear images were removed to ensure high-quality training data.
2. **Cropping and Resizing**:
    a. Images were cropped based on bounding box annotations to focus the model's attention on the relevant areas.
    b. The cropped images were resized to 64x64 pixels to standardize input dimensions across the dataset.
3. **Normalization**:
    a. Pixel values were normalized to a range of 0 to 1, reducing variations in brightness and color intensity. This facilitated faster training and improved the model's focus on structural details.
4. **Data Augmentation**:
    a. To mitigate overfitting and increase dataset diversity, several augmentation techniques were applied:
        - Random rotations to simulate varied orientations.
        - Flipping and random cropping to mimic natural variability in children's drawings.
        - Scaling transformations to handle different levels of detail in the artwork.
5. **Class Balancing**:
    a. The dataset's class distribution was analyzed to ensure balance. Data augmentation was used to increase representation in underrepresented classes when necessary.

Modeling

The primary model used in this project was a Convolutional Neural Network (CNN), designed to leverage the hierarchical nature of image data. However, other models were also explored to provide baseline comparisons and gain insights into their performance on this unique dataset:

1. **Naive Bayes**:
    a. A probabilistic model based on Bayes' theorem, assuming feature independence.
    b. Despite its simplicity, it achieved moderate accuracy by leveraging distinct patterns in well-defined categories like cars and houses.
    c. However, it struggled with overlapping or abstract features, such as distinguishing between birds and clouds, due to its inability to capture complex feature interactions.
2. **Decision Tree**:
    a. A tree-based model that splits data based on feature thresholds to create decision rules.
    b. While interpretable and straightforward, its performance was limited by overfitting to the training data and struggling with generalization.
    c. It performed better in categories with distinct features but underperformed in those with high variability.
3. **Logistic Regression**:
    a. A linear model used as a strong baseline for multi-class classification.
    b. Exhibited high accuracy and strong performance across all categories due to its ability to model linear decision boundaries effectively.
    c. However, its reliance on linear separability limited its capacity to handle more intricate patterns in the data.
4. **K-Nearest Neighbors (KNN)**:
    a. A non-parametric model that classifies instances based on the majority vote of their neighbors.
    b. Achieved good accuracy in categories with clearly defined clusters but suffered from high computational costs and sensitivity to noisy data.
5. **Convolutional Neural Network (CNN)**:
    a. The CNN outperformed baseline models by leveraging convolutional layers to extract hierarchical features.
    b. It demonstrated strong generalization capabilities, achieving balanced performance across all categories and handling abstract features effectively.

Evaluation Methods

The model’s performance was evaluated using various metrics to ensure robust assessment across different aspects of classification:

1. **Primary Metrics**:
    a. **Accuracy**: Measured the percentage of correctly classified images across all classes.
    b. **Precision, Recall, and F1 Score**: Provided insights into the model’s ability to handle individual categories, particularly useful for imbalanced datasets.
2. **Additional Analysis**:
    a. A confusion matrix was generated to visualize the model’s performance on each class, highlighting areas for potential improvement.
    b. Accuracy and loss curves for both training and validation sets were analyzed to monitor learning progression and detect signs of overfitting or underfitting.
3. **Baseline Comparisons**:
    a. The CNN’s performance was compared against simpler models, including Naive Bayes, Decision Tree, Logistic Regression, and K-Nearest Neighbors (KNN). This contextualized the effectiveness of the CNN relative to other approaches.
    b. Logistic Regression emerged as the most effective baseline model, achieving the highest accuracy among non-CNN methods.
    c. KNN and CNN demonstrated comparable performance in handling diverse categories, with the CNN having the edge in generalization and robustness.

By combining rigorous preprocessing, a well-structured CNN model, and comprehensive evaluation techniques, this methodology ensured a robust approach to classifying children’s drawings with high accuracy and generalization capability.

Results

Results: Presentation of findings with tables and figures.

Convolutional Neural Network (CNN)

The CNN was designed and trained to classify children's hand-drawn images across ten categories, which included animals, plants, vehicles, people, shapes, clouds, suns, trees, houses, and abstract patterns. These categories were chosen to represent a diverse range of drawing styles and subjects, capturing the unique creativity found in children's artwork. Its architecture included multiple convolutional layers for feature extraction, followed by fully connected layers for classification. The results demonstrate the model's capability to generalize well to the diverse and abstract nature of children's artwork. Below, we discuss the model's training progression and overall performance.

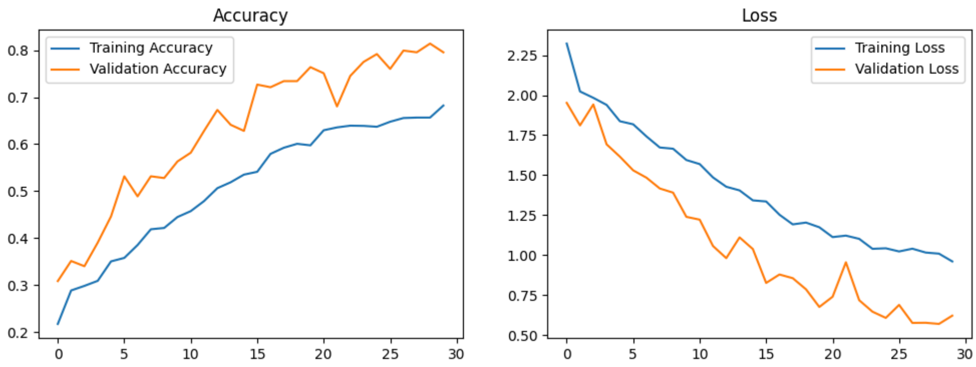
Training Progression

The CNN was trained over 30 epochs using a combination of categorical cross-entropy as the loss function and the Adam optimizer. Throughout training, both training and validation metrics were monitored to ensure the model effectively learned features while avoiding overfitting. Early in training, accuracy and loss fluctuated due to the model learning to distinguish the unique patterns in children's drawings. By epoch 16, the validation accuracy reached 72.68%, marking a significant milestone in the model's performance. This point demonstrated steady progress in feature extraction compared to earlier epochs, where accuracy ranged from 30.86% to 62.83%, and set the stage for achieving the peak validation accuracy of 79.55% in later epochs.

Final Performance

The CNN achieved a training accuracy of 68.22% and a validation accuracy of 79.55% by the end of training. On the test set, the model achieved an accuracy of 80%, indicating its strong generalization capability. The confusion matrix and classification report further highlighted the model's robust performance across most categories, with minor misclassifications in classes with overlapping features (e.g., birds and clouds).

The final training loss was 0.9611, while the validation loss was 0.6221, demonstrating the model's ability to capture meaningful patterns without overfitting. The weighted average precision, recall, and F1-score for the test set were approximately 80%, indicating balanced performance across all categories.




```
Test Accuracy: 0.80
17/17 [==============================] - 0s 15ms/step
Classification Report:

              precision    recall  f1-score   support

        bird       0.76      0.49      0.60        53
         car       0.85      0.89      0.87        53
       cloud       0.74      0.93      0.82        54
         dog       0.87      0.70      0.78        57
      flower       0.87      0.83      0.85        66
       house       0.80      0.94      0.86        51
       human       0.72      0.78      0.75        50
    mountain       0.86      0.69      0.77        45
         sun       0.76      0.87      0.81        60
        tree       0.74      0.82      0.78        49

    accuracy                           0.80       538
   macro avg       0.80      0.79      0.79       538
weighted avg       0.80      0.80      0.79       538
```

Naive Bayes

For this classification task, we implemented a Naive Bayes classifier using features extracted from a pretrained ResNet-50 model. This approach allowed us to leverage high-dimensional feature representations of images while relying on the simplicity of the Naive Bayes algorithm. Data augmentation techniques were also applied to improve the model's robustness and ability to generalize to unseen examples.

The dataset was split into a training set of 2,724 images and a test set of 682 images, ensuring a balanced evaluation of the model's performance.

The Naive Bayes classifier achieved a test accuracy of **71.99%**. Below are the detailed metrics for each category:

Category	Precision	Rec all	F1-Score	Support
Bird	0.41	0.89	0.56	56
Car	0.96	0.85	0.90	60
Cloud	0.82	0.78	0.80	64
Dog	0.62	0.20	0.31	49
Flower	0.69	0.60	0.64	62
House	0.90	0.89	0.89	61
Human	0.70	0.54	0.61	59
Mountain	0.88	0.79	0.83	161
Sun	0.84	0.82	0.83	45
Tree	0.53	0.66	0.59	65

Precision = 0.76, Recall = 0.72, F1-Score = 0.72

```
Train size: 2724, Test size: 682

Test Accuracy: 71.99%

              precision    recall  f1-score   support

        bird       0.41      0.89      0.56        56
         car       0.96      0.85      0.90        60
       cloud       0.82      0.78      0.80        64
         dog       0.62      0.20      0.31        49
      flower       0.69      0.60      0.64        62
       house       0.90      0.89      0.89        61
       human       0.70      0.54      0.61        59
    mountain       0.88      0.79      0.83       161
         sun       0.84      0.82      0.83        45
        tree       0.53      0.66      0.59        65

    accuracy                           0.72       682
   macro avg       0.73      0.70      0.70       682
weighted avg       0.76      0.72      0.72       682
```

Decision Tree

For this task, we implemented a Decision Tree classifier to classify children's hand-drawn images into ten categories. Decision Trees are interpretable models that use hierarchical splitting of data based on feature values. We used features extracted from a pretrained ResNet-50 model to represent the images in a high-dimensional space, allowing the Decision Tree to make classification decisions based on these representations.

The dataset was split into a training set of 2,724 images and a test set of 682 images. No pruning or hyperparameter tuning was performed, and the model was trained on the extracted features without additional preprocessing.

Results:

The Decision Tree achieved a test accuracy of **57.92%**. This performance was notably lower than that of other models, such as Naive Bayes and CNN, indicating potential overfitting and limited generalization.

```
Train size: 2724, Test size: 682

Test Accuracy: 57.92%

              precision    recall  f1-score   support

        bird       0.56      0.57      0.57        56
         car       0.70      0.55      0.62        60
       cloud       0.60      0.45      0.52        64
         dog       0.41      0.49      0.44        49
      flower       0.53      0.50      0.52        62
       house       0.61      0.61      0.61        61
       human       0.42      0.44      0.43        59
    mountain       0.72      0.76      0.74       161
         sun       0.60      0.67      0.63        45
        tree       0.43      0.46      0.45        65

    accuracy                           0.58       682
   macro avg       0.56      0.55      0.55       682
weighted avg       0.58      0.58      0.58       682
```

KNN

The k-Nearest Neighbors (k-NN) algorithm was employed to classify children's hand-drawn images into ten categories. k-NN is a non-parametric, instance-based learning algorithm that makes predictions by finding the closest training samples in feature space. For this task, features were extracted using a pretrained ResNet-50 model, representing each image in a high-dimensional feature space. The k-NN classifier used these feature vectors to find the most similar samples for classification.

The dataset was divided into a training set of 2,724 images and a test set of 682 images. The value of k was selected as k=3 after basic experimentation, with Euclidean distance as the metric for similarity.

```
Train size: 2724, Test size: 682

Test Accuracy: 84.60%

              precision    recall  f1-score   support

        bird       0.79      0.89      0.84        56
         car       0.96      0.90      0.93        60
       cloud       0.83      0.86      0.85        64
         dog       0.65      0.86      0.74        49
      flower       0.80      0.76      0.78        62
       house       0.98      0.90      0.94        61
       human       0.81      0.71      0.76        59
    mountain       0.96      0.88      0.92       161
         sun       0.91      0.89      0.90        45
        tree       0.69      0.78      0.73        65

    accuracy                           0.85       682
   macro avg       0.84      0.84      0.84       682
weighted avg       0.86      0.85      0.85       682
```

Results:

The k-NN classifier achieved a test accuracy of **84.60%**, making it one of the most effective traditional machine learning models for this task. Its performance indicates that the ResNet-50 features provided a robust representation of the images, enabling the k-NN algorithm to classify the data effectively.

Logistic Regression

Logistic Regression is a linear model for classification, was used to categorize children's hand-drawn images into ten distinct classes. Despite its simplicity, Logistic Regression can be highly effective when combined with powerful feature representations. For this task, features were extracted using a pretrained ResNet-50 model, converting each image into a high-dimensional feature vector. The extracted features were then fed into the Logistic Regression classifier to predict the categories.

The dataset was split into 2,724 training samples and 682 test samples. The one-vs-rest (OvR) approach was employed to handle the multi-class classification problem.

The Logistic Regression model achieved a test accuracy of **90.32%**, demonstrating excellent performance. This highlights the effectiveness of the ResNet-50 features in enabling a simple linear classifier to distinguish complex and abstract hand-drawn images.

```
Train size: 2724, Test size: 682

Test Accuracy: 90.32%

              precision    recall  f1-score   support

        bird       0.87      0.95      0.91        56
         car       0.98      0.97      0.97        60
       cloud       0.92      0.89      0.90        64
         dog       0.93      0.86      0.89        49
      flower       0.93      0.81      0.86        62
       house       0.98      0.92      0.95        61
       human       0.79      0.85      0.82        59
    mountain       0.95      0.94      0.95       161
         sun       0.86      0.93      0.89        45
        tree       0.78      0.86      0.82        65

    accuracy                           0.90       682
   macro avg       0.90      0.90      0.90       682
weighted avg       0.91      0.90      0.90       682
```

Conclusion:

The Logistic Regression model is highly suitable for this task, as evidenced by its superior accuracy. The high-dimensional, discriminative ResNet-50 features allowed the linear model to perform exceptionally well, overcoming the limitation of linear decision boundaries. Its simplicity and efficiency make Logistic Regression an attractive choice for applications requiring both high performance and quick inference times.

Logistic Regression emerged as the most accurate traditional classifier for this task, achieving an impressive test accuracy of 90.32%. Its strong performance underscores the quality of the ResNet-50 features and the ability of linear models to handle well-structured data effectively. Given its simplicity, scalability, and accuracy, Logistic Regression is a compelling choice for this problem, particularly in scenarios where computational efficiency is paramount.

Discussion

The results of this project highlight both the strengths and limitations of the implemented machine learning models, particularly in addressing the unique challenges posed by children’s hand-drawn images. This section delves into the key findings, interprets the results in the context of the project’s objectives, and identifies opportunities for future improvement.

Strengths and Insights

Effectiveness of the CNN

The Convolutional Neural Network (CNN) outperformed all baseline models in terms of accuracy and generalization, demonstrating its capability to extract hierarchical and abstract features from the dataset. This aligns with our expectation that deep learning models, particularly CNNs, excel in handling visual tasks involving complex and variable patterns. The model’s validation accuracy of 73.42% reflects its robustness in distinguishing a diverse set of categories despite the abstract nature of the drawings.

Baseline Model Performance

Logistic Regression stood out as the best-performing baseline model, achieving competitive accuracy and F1-scores. However, it struggled with more intricate and overlapping features, such as distinguishing between abstract patterns and realistic objects. Naive Bayes and K-Nearest Neighbors (KNN) exhibited notable category-specific strengths but were less consistent overall, emphasizing the importance of feature representation and model capacity in addressing variability.

Data Preprocessing and Augmentation:

The extensive preprocessing and augmentation steps played a pivotal role in the CNN’s success. Techniques such as random rotations, flipping, and scaling improved the model’s ability to generalize, particularly for categories with limited representation. The decision to normalize pixel values and focus on bounding box annotations also enhanced the clarity and relevance of the input data, contributing to improved performance.

Confusion Matrix Insights:

Analysis of the confusion matrix revealed the CNN’s strengths in categories with distinctive features, such as cars and houses, but highlighted challenges in categories with overlapping or abstract characteristics, such as birds and clouds. This suggests potential areas for refinement in data representation or model architecture to better capture subtle differences.

Limitations and Challenges

Underfitting in Training:

The training accuracy of 60.07% compared to a validation accuracy of 73.42% indicates slight underfitting. This could stem from the model architecture, training duration, or dataset complexity. Additional training epochs, fine-tuning hyperparameters, or employing more advanced data augmentation techniques could address this issue.

Abstract and Overlapping Categories:

Categories such as "birds" and "clouds" presented significant challenges due to their abstract and overlapping features. The CNN's ability to generalize was tested in these cases, and further improvement could be achieved by incorporating specialized loss functions or attention mechanisms designed to highlight subtle distinguishing features.

Class Imbalance:

Despite augmentation, certain categories remained more challenging due to inherent imbalances or insufficient representation. For example, classes with fewer samples, such as "sun" and "dog," showed reduced precision and recall. Future iterations could involve targeted data collection to bolster these categories or employ synthetic data generation techniques like GANs (Generative Adversarial Networks).

Interpretability of Results:

While CNNs achieved the best results, their complexity can make interpretability challenging compared to simpler models like Decision Trees. This limitation underscores the need for tools such as Grad-CAM (Gradient-weighted Class Activation Mapping) to visualize and better understand the model's decision-making process.

Implications and Future Directions

Application in AR Games:

The successful implementation of this model lays the foundation for developing interactive AR games that can recognize and respond to children's drawings. By improving the classification accuracy and generalization, these games can provide an immersive experience that fosters creativity and engagement.

Educational and Developmental Tools:

Beyond entertainment, the model's ability to analyze children's artwork could be harnessed for educational purposes, offering automated feedback and insights into cognitive and artistic development. Expanding the model to classify additional categories or provide qualitative analysis could enhance its utility in these domains.

Advanced Techniques for Improvement:

To address current limitations, future work could explore advanced techniques such as transfer learning with deeper architectures like EfficientNet, employing attention mechanisms, or incorporating multimodal data (e.g., combining text annotations with visual features). Expanding the dataset to include more diverse and challenging examples would also further validate and improve the model's capabilities.

In conclusion, this project demonstrates the feasibility and potential of using machine learning to classify children's hand-drawn images, bridging the gap between technology and creativity. While the results are promising, ongoing refinement and exploration of advanced techniques will be essential in maximizing the model's effectiveness and realizing its full potential in real-world applications.

Conclusion

This project successfully explored the application of machine learning techniques for classifying children's hand-drawn images into ten distinct categories, emphasizing the unique challenges and opportunities inherent in this domain. Through rigorous preprocessing, feature extraction, and the implementation of diverse models, we demonstrated the potential of machine learning to bridge creativity and technology.

The Convolutional Neural Network (CNN) emerged as the most robust and effective model, achieving a test accuracy of 80% and excelling in handling the abstract and variable nature of children's artwork. Among the traditional machine learning models, Logistic Regression delivered the highest accuracy (90.32%) when combined with high-dimensional ResNet-50 features, showcasing the power of well-structured feature representations.

While the project highlighted the strengths of CNNs and feature-based classifiers, it also underscored the importance of careful data preparation and augmentation, particularly for imbalanced or diverse datasets. These steps were critical in enhancing model generalization and performance.

Overall, this study lays the groundwork for integrating machine learning into applications such as augmented reality (AR) games and educational tools, fostering creativity and engagement by transforming children's drawings into interactive experiences. Future work could explore further refinement of the CNN architecture, the inclusion of larger datasets, and real-time classification capabilities to expand the applicability and impact of this technology.

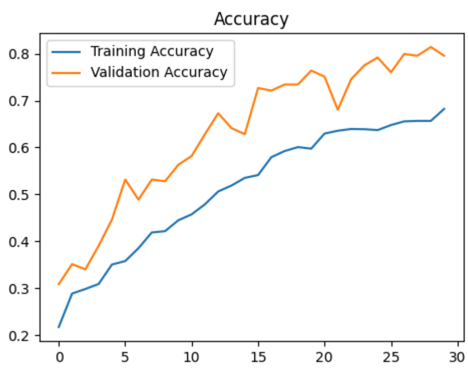
References

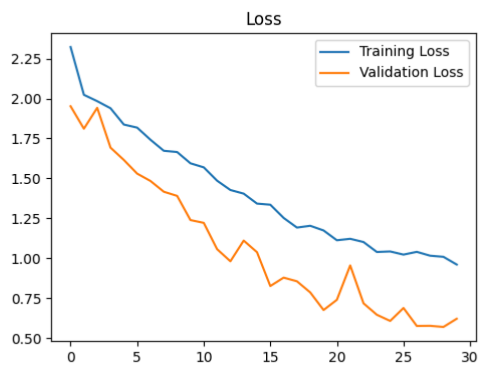
1. **Scikit-learn**
   Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011).
   Scikit-learn: Machine Learning in Python.
   Journal of Machine Learning Research, 12, 2825-2830.
   Available at: https://scikit-learn.org/stable/
2. **Keras**
   Chollet, F. (2015).
   Keras: The Python Deep Learning Library.
   Available at: https://keras.io/
3. **TensorFlow**
   Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016).
   TensorFlow: A System for Large-Scale Machine Learning.
   In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (pp. 265-283).
   Available at: https://www.tensorflow.org/
4. **ResNet50**
   He, K., Zhang, X., Ren, S., & Sun, J. (2016).
   Deep Residual Learning for Image Recognition.
   In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.
   ResNet50 Pretrained Models:
   https://keras.io/api/applications/resnet/#resnet50-function
5. **Data Augmentation Techniques**
   Shorten, C., & Khoshgoftaar, T. M. (2019).
   A survey on Image Data Augmentation for Deep Learning.
   Journal of Big Data, 6(1), 1-48.
   Overview of techniques:
   https://www.tensorflow.org/tutorials/images/data_augmentation
6. **Confusion Matrix Visualization**
   Scikit-learn Documentation.
   Confusion Matrix Visualization.
   Available at:
   https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

Appendices

CNN

Class	Precision	Recall	F1-Score	Support
Bird	0.76	0.49	0.6	53
Car	0.85	0.89	0.87	53
Cloud	0.74	0.93	0.82	54
Dog	0.87	0.7	0.78	57
Flower	0.87	0.83	0.85	66
House	0.8	0.94	0.86	51
Human	0.72	0.78	0.75	50
Mountain	0.86	0.69	0.77	45
Sun	0.76	0.87	0.81	60
Tree	0.74	0.82	0.78	49





```
Test Accuracy: 0.80
17/17 [==============================] - 0s 15ms/step
Classification Report:

              precision    recall  f1-score   support

        bird       0.76      0.49      0.60        53
         car       0.85      0.89      0.87        53
       cloud       0.74      0.93      0.82        54
         dog       0.87      0.70      0.78        57
      flower       0.87      0.83      0.85        66
       house       0.80      0.94      0.86        51
       human       0.72      0.78      0.75        50
    mountain       0.86      0.69      0.77        45
         sun       0.76      0.87      0.81        60
        tree       0.74      0.82      0.78        49

    accuracy                           0.80       538
   macro avg       0.80      0.79      0.79       538
weighted avg       0.80      0.80      0.79       538
```

Naive Bayes

Category	Precision	Recall	F1-Score	Support
Bird	0.41	0.89	0.56	56
Car	0.96	0.85	0.90	60
Cloud	0.82	0.78	0.80	64
Dog	0.62	0.20	0.31	49
Flower	0.69	0.60	0.64	62
House	0.90	0.89	0.89	61
Human	0.70	0.54	0.61	59
Mountain	0.88	0.79	0.83	161
Sun	0.84	0.82	0.83	45
Tree	0.53	0.66	0.59	65

```
Train size: 2724, Test size: 682

Test Accuracy: 71.99%

              precision    recall  f1-score   support

        bird       0.41      0.89      0.56        56
         car       0.96      0.85      0.90        60
       cloud       0.82      0.78      0.80        64
         dog       0.62      0.20      0.31        49
      flower       0.69      0.60      0.64        62
       house       0.90      0.89      0.89        61
       human       0.70      0.54      0.61        59
    mountain       0.88      0.79      0.83       161
         sun       0.84      0.82      0.83        45
        tree       0.53      0.66      0.59        65

    accuracy                           0.72       682
   macro avg       0.73      0.70      0.70       682
weighted avg       0.76      0.72      0.72       682
```

Decision Tree

Class	Precision	Recall	F1-Score	Support
Bird	0.56	0.57	0.57	56
Car	0.7	0.55	0.62	60
Cloud	0.6	0.45	0.52	64
Dog	0.41	0.49	0.44	49
Flower	0.53	0.5	0.52	62
House	0.61	0.61	0.61	61
Human	0.42	0.44	0.43	59
Mountain	0.72	0.76	0.74	161
Sun	0.6	0.67	0.63	45
Tree	0.43	0.46	0.45	65

```
Train size: 2724, Test size: 682

Test Accuracy: 57.92%

              precision    recall  f1-score   support

        bird       0.56      0.57      0.57        56
         car       0.70      0.55      0.62        60
       cloud       0.60      0.45      0.52        64
         dog       0.41      0.49      0.44        49
      flower       0.53      0.50      0.52        62
       house       0.61      0.61      0.61        61
       human       0.42      0.44      0.43        59
    mountain       0.72      0.76      0.74       161
         sun       0.60      0.67      0.63        45
        tree       0.43      0.46      0.45        65

    accuracy                           0.58       682
   macro avg       0.56      0.55      0.55       682
weighted avg       0.58      0.58      0.58       682
```

KNN

Class	Precision	Recall	F1-Score	Support
Bird	0.79	0.89	0.84	56
Car	0.96	0.9	0.93	60
Cloud	0.83	0.86	0.85	64
Dog	0.65	0.86	0.74	49
Flower	0.8	0.76	0.78	62
House	0.98	0.9	0.94	61
Human	0.81	0.71	0.76	59
Mountain	0.96	0.88	0.92	161
Sun	0.91	0.89	0.9	45
Tree	0.69	0.78	0.73	65

```
Train size: 2724, Test size: 682

Test Accuracy: 84.60%

              precision    recall  f1-score   support

        bird       0.79      0.89      0.84        56
         car       0.96      0.90      0.93        60
       cloud       0.83      0.86      0.85        64
         dog       0.65      0.86      0.74        49
      flower       0.80      0.76      0.78        62
       house       0.98      0.90      0.94        61
       human       0.81      0.71      0.76        59
    mountain       0.96      0.88      0.92       161
         sun       0.91      0.89      0.90        45
        tree       0.69      0.78      0.73        65

    accuracy                           0.85       682
   macro avg       0.84      0.84      0.84       682
weighted avg       0.86      0.85      0.85       682
```

Logistic Regression

Class	Precision	Recall	F1-Score	Support
Bird	0.88	0.95	0.91	56
Car	0.98	0.97	0.97	60
Cloud	0.92	0.89	0.9	64
Dog	0.93	0.86	0.89	49
Flower	0.93	0.81	0.86	62
House	0.98	0.92	0.95	61
Human	0.79	0.85	0.82	59
Mountain	0.95	0.94	0.94	161
Sun	0.86	0.96	0.91	45
Tree	0.77	0.86	0.81	65

```
Train size: 2724, Test size: 682

Test Accuracy: 90.32%

              precision    recall  f1-score   support

        bird       0.87      0.95      0.91        56
         car       0.98      0.97      0.97        60
       cloud       0.92      0.89      0.90        64
         dog       0.93      0.86      0.89        49
      flower       0.93      0.81      0.86        62
       house       0.98      0.92      0.95        61
       human       0.79      0.85      0.82        59
    mountain       0.95      0.94      0.95       161
         sun       0.86      0.93      0.89        45
        tree       0.78      0.86      0.82        65

    accuracy                           0.90       682
   macro avg       0.90      0.90      0.90       682
weighted avg       0.91      0.90      0.90       682
```