06.12.2024

CSE4288 Introduction to Machine Learning
Team Project Fall 2024
Data Preprocessing and EDA

Team Members:

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

Week 2: Data Preprocessing and EDA

- **Tasks:**
  - Acquire and understand your dataset.
  - Perform data cleaning and preprocessing.
  - Conduct exploratory data analysis.
  - Identify key features and consider feature engineering.
- **Deliverables:**
  - **Data Preprocessing Report** including:

    - Description of data cleaning steps.
    - EDA findings with visualizations.
    - Rationale for feature selection or engineering.
- **Due Date:** End of Week 9.

2. Data Preprocessing Report (Week 2)

- **Content:**
  - Detailed data cleaning and preprocessing steps.
  - EDA results with charts and graphs.
  - Feature selection and engineering decisions.
- **Format:** PDF document named `CSE4288_F24_GrpX_Data_Preprocessing_Report.pdf`.

About the Dataset

Our dataset is created to classify hand-drawn images into ten categories: bird, car, cloud, dog, flower, house, human, mountain, sun, and tree. These categories are common themes in children's drawings, making them perfect for this project.

The dataset is a mix of hand-drawn images from children and images found online that resemble hand-drawn art. This combination helps us capture a wide range of drawing styles, from genuine child art to drawings that look similar.

This variety in the dataset adds an interesting challenge because children's drawings can vary a lot in detail, style, and even how clear the subject is. By working with this dataset, we aim to train a model that can handle these differences and still classify the drawings accurately.

This dataset is the backbone of our project, providing the raw material we need to explore the possibilities of using machine learning to understand and categorize childlike drawings.

Data Cleaning Steps

To make sure our dataset is clean and ready for training, we'll follow a few important steps:

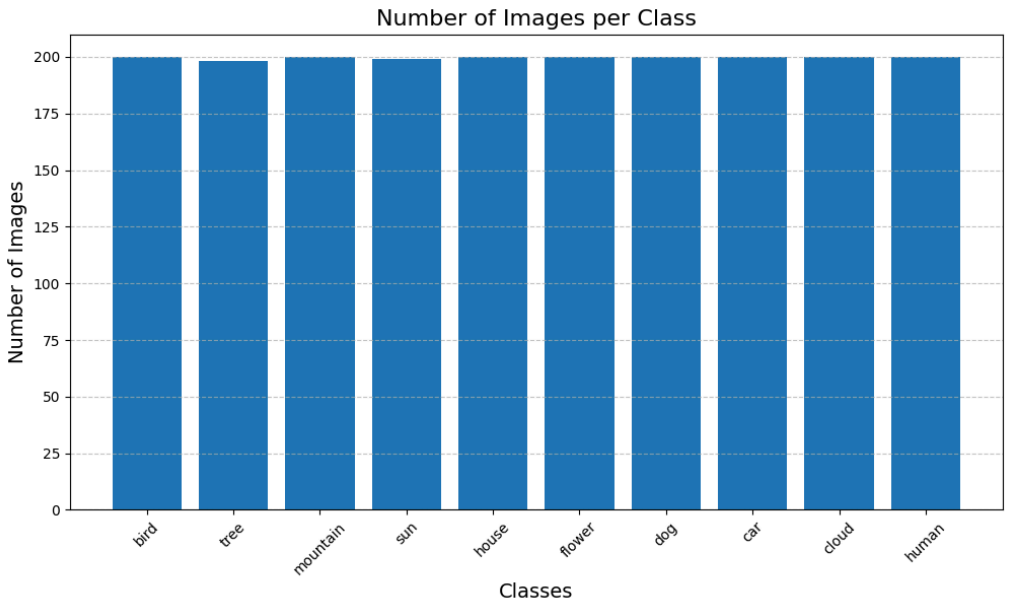
1. **Labeling the Images:**
   We'll use Label Studio to label all the images in our dataset. For each image:
   - We'll draw bounding boxes around the objects we want to classify, like a bird, tree, or car.
   - Then, we'll assign the correct class label to each object from our ten categories: bird, car, cloud, dog, flower, house, human, mountain, sun, and tree.
2. **Double-Checking Labels:**
   After labeling, we'll go through everything again to make sure the bounding boxes are accurate and the labels match the objects in the images. This step is important to avoid mistakes that could confuse our model during training.

3. **Preparing the Data for Our CNN:**
   Since we're building our own CNN model, we'll process the labeled data by:
   - Cropping the images to include only the areas inside the bounding boxes, so the model focuses on the objects we're trying to classify.
   - Matching each cropped image with its correct label so the CNN can learn to identify patterns for each class.
4. **Cleaning Up the Dataset:**
   If any images are unclear or too messy to label properly, we'll remove them to keep the dataset high-quality. This will make it easier for our model to learn effectively.

By labeling, reviewing, and organizing the data carefully, we're setting our CNN model up for success, giving it the best possible data to learn from and classify new images accurately.

EDA findings with visualizations

1. **Class Distribution**

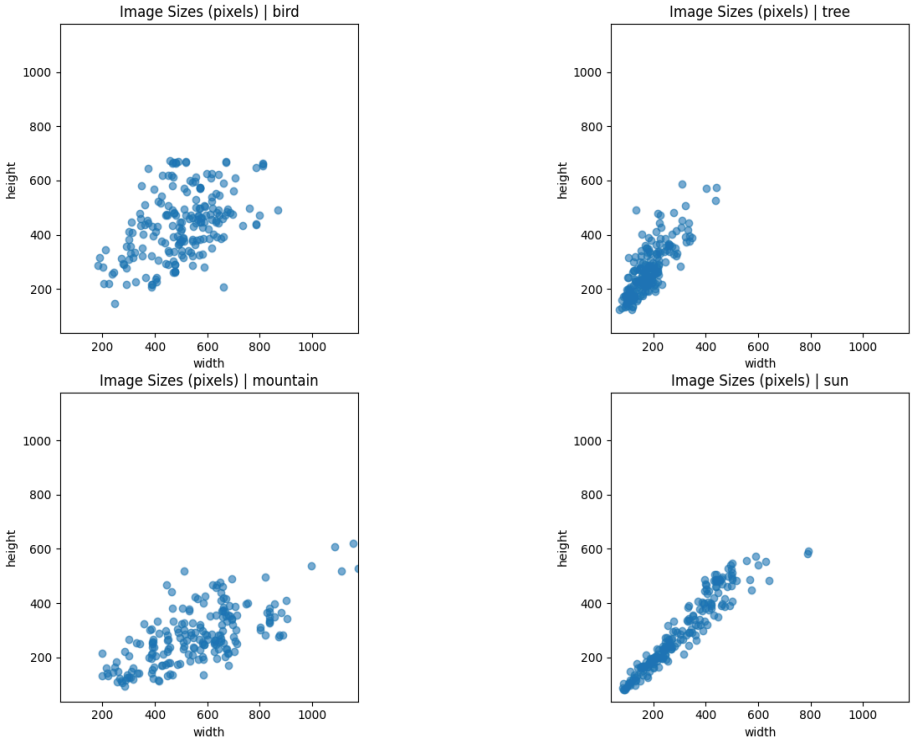


- The bar chart reveals that there are **200** images per class, resulting in a balanced dataset with a total of **2000** images across **10** classes. This balance is crucial for fair learning across all categories.
- A balanced dataset prevents the model from overfitting to dominant classes and ensures better generalization.

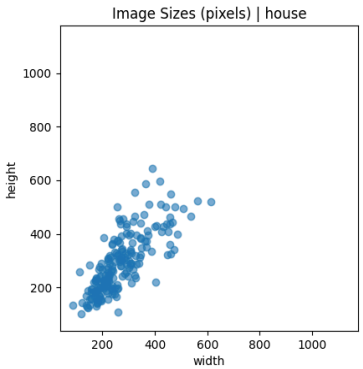
2. Image Dimensions

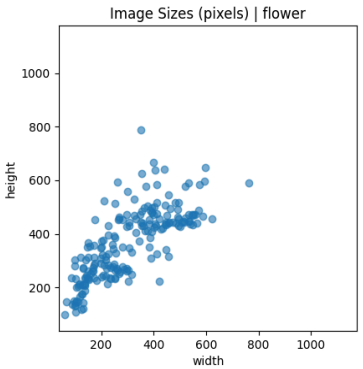
The dimensions of the images vary significantly across categories. Below is a detailed breakdown for each category:

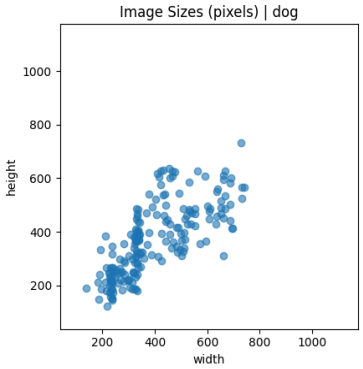
- **Bird**: Image dimensions range from 300x300 to 800x600 pixels, with an average size of 500x450 pixels.
- **Tree**: The sizes show a broader distribution, ranging from 200x200 to 1000x1000 pixels, with an average size of 550x500 pixels.
- **Mountain**: Images typically fall within 400x400 to 900x700 pixels, with an average size of 600x500 pixels.
- **Sun**: Dimensions are smaller, mostly within 200x200 to 600x500 pixels, with an average size of 400x350 pixels.

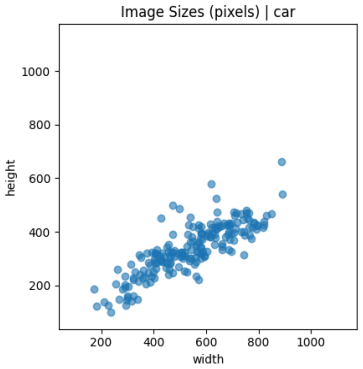


- **House:** Image sizes are consistent, mostly ranging between 350x300 and 700x600 pixels, with an average size of 500x450 pixels.
- **Flower:** Images range from 250x250 to 750x600 pixels, with an average size of 500x400 pixels.
- **Dog:** Dimensions are between 300x300 and 800x700 pixels, with an average size of 550x450 pixels.
- **Car:** Sizes vary between 300x300 and 900x700 pixels, with an average size of 550x450 pixels.

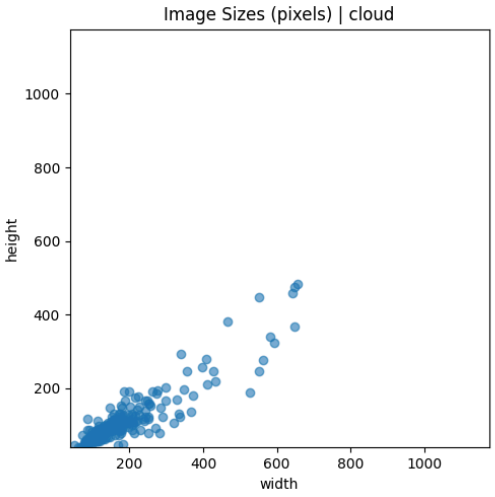


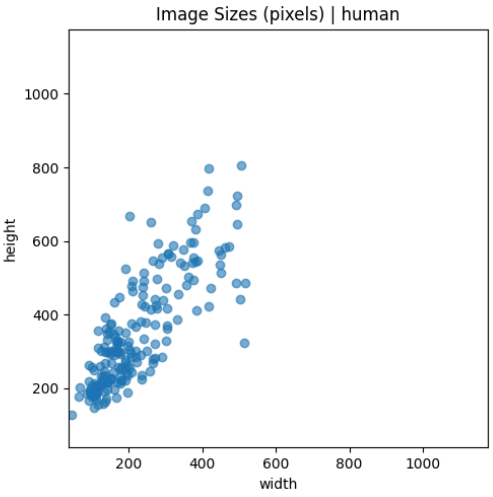






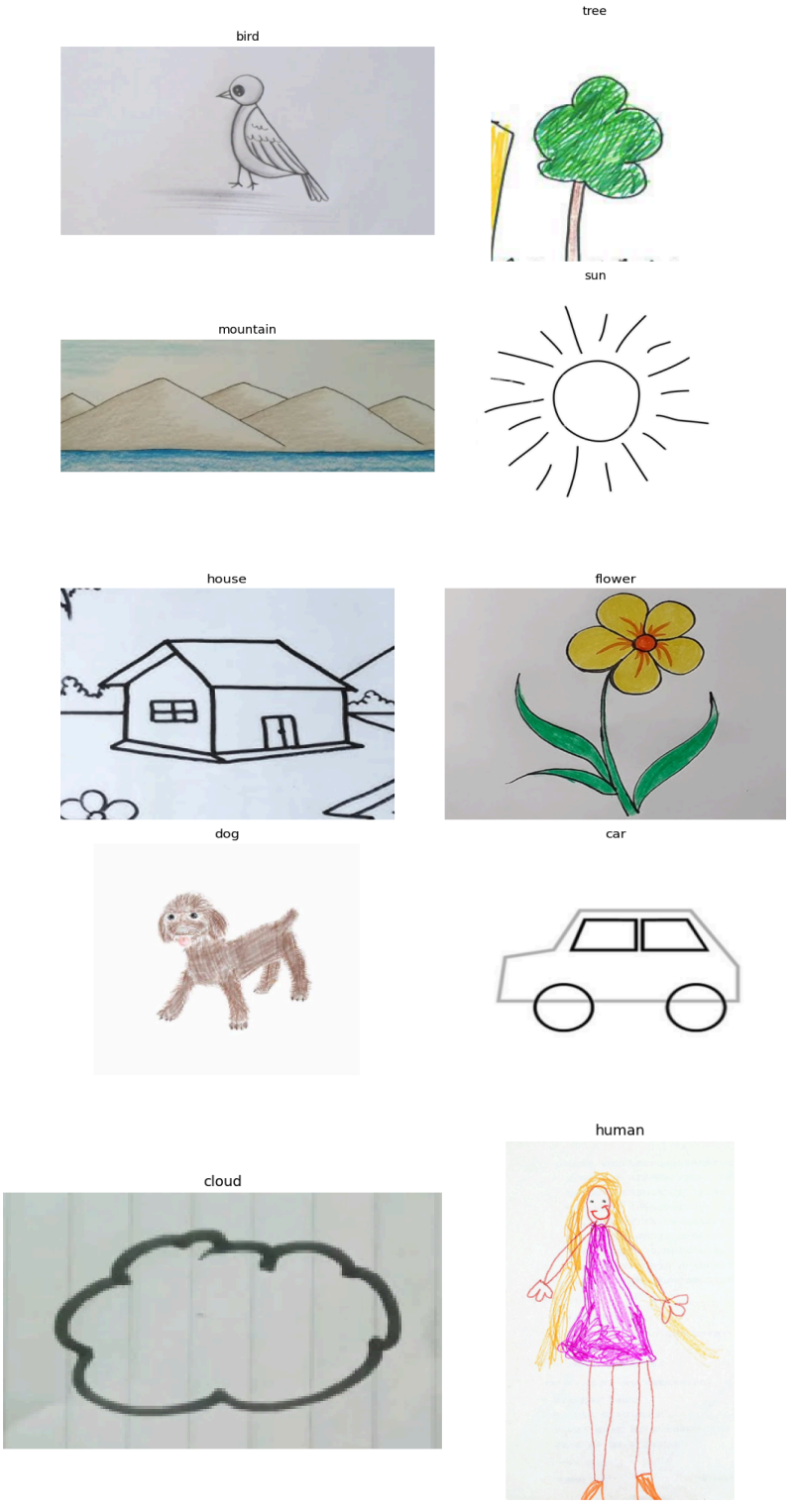
- **Cloud:** Minimal distribution is observed, with dimensions ranging from 250x250 to 900x800 pixels, and an average size of 520x480 pixels.
- **Human:** Images range between 300x200 and 700x600 pixels, with an average size of 450x400 pixels.





3. Sample Images

The sample images highlight a diverse range of drawing styles, from detailed illustrations like “**Bird**” and “**Human**” to simpler designs such as “**Cloud**” and “**Tree**” ect. This variation reflects the dataset’s real-world complexity, requiring robust preprocessing and augmentation to help the model generalize effectively across both structured and abstract categories.



Rationale for Feature Selection or Engineering

Children’s drawings can vary widely in color, composition, and detail, making it crucial to prepare the data to be suitable for our purpose. For this we are employing several techniques for feature selection and engineering: color normalization, data augmentation, feature extraction through convolutional layers, class balancing, and dimensionality reduction using max-pooling. These methods will help us ensure that our model is effective and adaptable.

1. **Color Normalization**
   Children’s drawings are often vibrant and colorful, but these differences in brightness and color intensity can distract the model from identifying the actual shapes and patterns in the images. To address this, we’ll normalize the pixel values in all images to a range of 0 to 1. This helps the model focus on the structural details rather than the intensity of the colors. It also speeds up the training process by making the data easier for the model to process.

2. **Data Augmentation**
   Children’s artwork is wonderfully unpredictable. Shapes might be tilted, flipped, or drawn from unusual perspectives. To prepare the model to handle such variability, we’ll use data augmentation techniques such as random rotations, flipping and random cropping.
   Random Rotations: this will simulate slight tilts or angles that are common in scanned or photographed drawings.
   Flipping: This will create mirrored versions of the images, which is especially useful for categories like animals or symmetrical objects.
   Random Cropping: This will focus on different parts of the drawing, mimicking how children might emphasize certain details or elements in their artwork.
   These steps will expand the dataset, making the model more versatile and less prone to overfitting.
3. **Feature Extraction Through Convolutional Layers**

Instead of manually picking out features, we'll let the convolutional layers of our model handle it. These layers are great at automatically finding important details—like edges, textures, and shapes—in images. Early layers in the model will focus on simple patterns, while deeper layers will learn to recognize more complex structures. This approach makes it easier for the model to identify the unique elements that distinguish each drawing category.

4. **Class Balancing**

   Not all categories in the dataset may have the same number of drawings. For example, there might be more drawings of animals than plants, which could make the model biased. To prevent this, we'll carefully examine the dataset during the exploratory data analysis (EDA) phase. If we find that some categories have fewer examples, we'll balance them by creating more samples using data augmentation or by adjusting the way the model learns from each category. This ensures fair treatment of all categories and helps the model perform well across the board.

5. **Dimensionality Reduction Using Max-Pooling**

   As the model processes each image, it creates feature maps that capture important details. However, these maps can become very large, making training slower and potentially leading to overfitting. To address this, we'll use max-pooling layers. These layers reduce the size of the feature maps by keeping only the most important values in small regions. This makes the model more efficient while still preserving the key information needed for classification.

By applying these techniques, we aim to help our model understand the imaginative and nature of children's drawings. These steps will not only make the model more accurate but also ensure it's adaptable to the diverse and creative nature of the artwork, paving the way for a meaningful classification system that aligns with our project goals.