Data Collection and Preprocessing Phase

Date

24 July 2024

Team ID

SWUID20240034367

Project Title

Anemia-Sense-Leveraging-Machine-Learning-For-Precise-Anemia-Recognitions-using-python

Maximum Marks

6 Marks

Preprocessing Handle Missing Values: Impute or remove missing values.

Detect and Handle Outliers: Identify and address outliers.

Normalize and Standardize: Scale numerical variables.

Encode Categorical Variables: Convert categorical variables to numerical representations.

Feature Engineering: Create or select relevant features.

Split Data: Divide data into training and testing sets.

Section

Description

The dataset for this project was sourced from Kaggle, a well-known platform for datasets and machine learning competitions. The specific dataset used, provided by Biswa Ranjan Rao, consists of 1421 samples with six attributes: gender, hemoglobin levels, mean corpuscular hemoglobin (MCH), mean corpuscular hemoglobin concentration (MCHC), mean corpuscular volume (MCV), and the result (binary classification of anemia status). This dataset was chosen for its comprehensiveness and relevance to the task of anemia detection, as it includes key hematological parameters commonly used by clinicians to diagnose anemia. Upon acquiring the dataset, it was crucial to validate its integrity. This involved verifying that the data was complete, free from corruption, and consistent with the metadata description. The memory size of the dataset, calculated at 66.7 MB, was noted to ensure that the data handling processes could be managed efficiently within the available computational resources.

Data Overview

Data Overview for Anemia-Sense

Data Sources:

- Electronic Health Records (EHRs)
- Public Health Databases
- Research Studies

Data Types:

- **Numerical:** Age, hemoglobin levels, hematocrit, MCV, etc.
- **Categorical:** Gender, ethnicity, medical history, diagnoses.

Data Quality:

- **Completeness:** Missing values, completeness rate.
- **Accuracy:** Consistency checks, data validation.
- **Timeliness:** Data age, update frequency.
- **Relevance:** Alignment with project objectives.

Feature selection-

Feature selection is a critical step that involves identifying the most important attributes that contribute to the prediction of anemia. In this project, multiple techniques were employed:

• **Correlation Analysis:** This method was used to evaluate the strength of the relationship between each attribute and the target variable. Attributes with strong correlations were considered for inclusion in the model.

• **SelectKBest:** This technique was applied to rank features based on their relevance to the target variable, selecting the top k features that provided the most predictive power.

• **Extra Trees Classifier**: An ensemble method that ranks features based on their importance was also used. This technique helped in identifying features that might not have strong individual correlations with the target but were still important when considered in combination with other features.

The goal of feature selection was to reduce the dimensionality of the dataset, improve model performance, and prevent overfitting by eliminating irrelevant or redundant features.

Normalisation

Normalization is a crucial preprocessing step for the Anemia-Sense project to ensure that features with different scales are treated fairly by the machine learning algorithms.

Data Augmentation

Data augmentation techniques such as horizontal flipping, random rotations, zooming, and shifting will be applied. This will artificiallyincrease the size of the dataset and improve the model's ability to generalize by introducing variability in the training images.

Data Preprocessing

clean, transform, and normalize the data to ensure its suitability for machine learning.

Image Cropping

Images will be cropped to focus on the dog, removing any background clutter that might be present. This step is essential for ensuring that the model's attention is focused on the object of interest.

Batch Normalization

During the neural network training, batch normalization will be applied to the input of each layer. This technique helps in accelerating training and improving the stability of the model bynormalizing the input to each layer.

Data Preprocessing Code Screenshots

Loading Data

```
from tensorflow.keras.preprocessing.image import load_img,img_to_array

image = load_img('path/to/image.jpg')image_array =
img_to_array(image)
```

Resizing

```
from tensorflow.keras.preprocessing.image import load_img,img_to_array

image = load_img('path/to/image.jpg', target_size=(224, 224))image_array =
img_to_array(image)
```

Normalization

```
normalized_image = image_array / 255.0
```

Data Augmentation

```
from tensorflow.keras.preprocessing.image importImageDataGenerator


    datagen = ImageDataGenerator(rotation_range=20,
    width_shift_range=0.2, height_shift_range=0.2,
    shear_range=0.2, zoom_range=0.2,
```

```
    horizontal_flip=True,fill_mode='nearest'
)
augmented_image = datagen.random_transform(image_array)
```

Denoising

```
import cv2 denoised_image =
cv2.fastNlMeansDenoisingColored(image_array.astype('uint8'), None, 10, 10, 7, 21)
```

Edge Detection

```
import cv2
gray_image = cv2.cvtColor(image_array.astype('uint8'),cv2.COLOR_RGB2GRAY)
edges = cv2.Canny(gray_image, 100, 200)
```

Color Space Conversion

```
import cv2
hsv_image = cv2.cvtColor(image_array.astype('uint8'),cv2.COLOR_RGB2HSV)
```

Image Cropping

```
start_x, start_y = 50, 50 # top-left corner of the crop width, height = 150, 150  # width
and height of the crop

cropped_image = image_array[start_y:start_y+height,start_x:start_x+width]
```