**Report for ML model**

**Automatic Traffic Sign Recognition**

**Zhumanov Yernur and Utumov Ramazan**

**BDA-2204**

Date: 19.11.2024

**Overview**

This report outlines the development and evaluation of an image classification model aimed at categorizing images into 58 distinct classes. The model was trained on a dataset of annotated images and was evaluated using various metrics to assess its performance. The main focus was addressing class imbalance and building a robust architecture for generalization.

Two models were developed:

1. **Baseline Model**: A simple CNN to serve as a benchmark for performance.
2. **Enhanced Model**: An advanced architecture incorporating additional regularization and layers for better feature extraction.

Both models were trained on a balanced dataset and evaluated across multiple metrics to identify strengths and limitations.

## **Data Preparation**

### **Dataset Description**

* **Number of Classes**: 58
* **Total Samples**: The dataset contains thousands of images with bounding box annotations specifying the regions of interest.
* **Annotation Details**:
  + The dataset includes a CSV file containing image file names, bounding box coordinates, and class labels.
  + Images are cropped using bounding box coordinates before resizing.

### **Preprocessing Steps**

1. **Cropping**: Images are cropped to focus on the bounding boxes specified in the annotations.
2. **Resizing**: All cropped images are resized to 128x128 pixels to standardize input dimensions.
3. **Normalization**: Pixel values are scaled to the range [0, 1] for improved convergence during training.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

### **Data Splitting**

* The dataset was split into training, validation, and test sets as follows:
  + **Training Set**: 4,314 samples (70% of data) used for model training.
  + **Validation Set**: 925 samples (15%) used for hyperparameter tuning and performance monitoring.
  + **Test Set**: 925 samples (15%) held out for final evaluation.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

### **Class Imbalance and Weights**

The dataset exhibits significant class imbalance, with some categories being heavily underrepresented. To address this:

* **Class Weights**: Computed based on the inverse frequency of each class to penalize misclassification of rare categories during training.

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Автоматически созданное описание

## **Baseline Model**

### **Architecture**

The baseline model serves as a foundational approach, implementing a simple CNN architecture. It focuses on extracting features with basic convolutional layers, followed by dense layers for classification:

* **Convolutional Layers**: Extract spatial features from images.
* **Pooling Layers**: Reduce spatial dimensions, preventing overfitting.
* **Dense Layers**: Convert features into predictions for 58 classes.

### **Training Setup**

* **Optimizer**: Adam with an initial learning rate of 0.001, chosen for its adaptive learning rate capabilities.
* **Loss Function**: Categorical Crossentropy, suitable for multi-class classification tasks.
* **Batch Size**: 32, balancing memory efficiency and convergence speed.
* **Augmentation**: Basic augmentations such as horizontal flips and rotations were applied to improve generalization.

### **Performance**

* **Strengths**:
  + The model performed well on frequently occurring classes.
* **Limitations**:
  + Struggled with underrepresented categories, highlighting the need for class weighting and enhanced feature extraction.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

## **Enhanced Model**

### **Key Improvements**

The enhanced model builds upon the baseline, introducing modifications to better capture complex patterns and handle imbalanced data:

1. **Additional Layers**: More convolutional layers were added to deepen the feature extraction process.
2. **Batch Normalization**: Introduced after convolutional layers to stabilize and accelerate training.
3. **Dropout Regularization**: Applied with a rate of 0.5 to mitigate overfitting.
4. **Dense Layer**: Expanded to 128 neurons with ReLU activation for learning richer feature representations.

### **Training Setup**

* The optimizer, learning rate, and batch size remained consistent with the baseline.
* The model leveraged class weights during training to emphasize rare categories.

### **Observations**

* **Improved Generalization**:
  + Rare classes showed better recall due to class weighting.
  + Additional layers allowed the model to detect finer details.
* **Challenges**:
  + Despite regularization, overfitting began appearing in later epochs.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

## **Evaluation Metrics**

Both models were evaluated using standard classification metrics:

* **Accuracy**: Percentage of correctly classified samples across all classes.
* **Precision**: Ability to avoid false positives in predictions.
* **Recall**: Ability to correctly identify all samples of each class.
* **F1-Score**: Harmonic mean of precision and recall, balancing their trade-offs.

**Enhanced Model**:

* Demonstrated higher precision and recall for underrepresented classes compared to the baseline.
* Achieved a notable increase in overall F1-score, indicating better performance balance.

Изображение выглядит как текст, снимок экрана, меню, программное обеспечение

Автоматически созданное описание

## **Deployment**

The final enhanced model was deployed using an interactive interface, providing user-friendly functionality for image classification.

### **Deployment Features**

1. **File Upload**: Users can upload images directly for real-time classification.
2. **Prediction Display**:
   * The interface returns the predicted class and confidence score for each image.
3. **Scalability**: The deployment supports batch processing of multiple images simultaneously.

### **Use Cases**

* Document classification in archival systems.
* Automated labeling for large-scale image datasets.

**Github link:** [**https://github.com/Roma313/Automatic-Traffic-Sign-Recognition**](https://github.com/Roma313/Automatic-Traffic-Sign-Recognition)