Garbage Classification

Calliope Bandera

1 Introduction

In a world always more polluted where tons of garbage is produced everyday, recycling has become fundamental. Recycling is one of the few ways we can help the environment but manually sorting waste can be a challenging endeavour. By automating the classification process through image analysis, this project seeks to enhance accuracy and alleviate the burden on humans, promoting a more effective recycling system.

The classification task involves 12 distinct classes commonly found in household waste: paper, cardboard, biological, metal, plastic, green-glass, brown-glass, white-glass, clothes, shoes, batteries, and trash. Notably, glass is further categorized into multiple classes to facilitate a more detailed and accurate classification. The dataset used for training and evaluation was taken from Kaggle, where it was created combining multiple existing datasets and augmenting them through web scraping. [1]

2 Methods/Case Study

2.1 First considerations

In the initial stages of this project, multiple approaches were considered to tackle the task of household garbage classification.

Linear regression was initially considered for garbage classification. However, it was quickly discarded as it might have not been very well-suited for this task. This classification task involves distinguishing between multiple classes, each with non-linear relationships in image feature spaces. Given that linear regression is designed for predicting continuous values, it would have struggled to capture the complex and non-linear patterns in the images.

Another option was the use of a Convolutional Neural Network (CNN) architecture without using pre-trained weights from ImageNet. Training a CNN from scratch could have been a valid approach, but the limited size of the dataset posed a significant challenge. Training a CNN without pre-trained weights on a small dataset might lead to overfitting and bad performance.

2.2 Approach based on State-of-the-art

In the final approach, ImageNet was selected as the basis for the project. ImageNet is a comprehensive image database extensively employed for training deep neural networks. The strategy involved using a pre-trained model on ImageNet, and specifically fine-tuning its last layer for garbage classification. This choice is based on the principles of transfer learning.

Transfer learning, and specifically fine-tuning, allows to use the knowledge captured by ImageNet's extensive training on diverse image data. The pre-trained model has learned features that are generally applicable to various image recognition tasks. By fine-tuning the model on the specific garbage dataset, these general features can be adapted to the task of garbage classification, leveraging the pre-trained model's ability to recognize basic visual patterns.

2.3 Model

For the task, a convolutional neural network (CNN) architecture was chosen due to its effectiveness in image-related tasks. The specific model that was used was:

- **MobileNetV2 Base Model**:

  The initial layer employs the MobileNetV2 architecture with pre-trained weights from ImageNet. It is a lightweight and computationally efficient neural network architecture.

- **GlobalMaxPooling2D Layer**:

  Following MobileNetV2, a GlobalMaxPooling2D layer is introduced for spatial downsampling. The maximum value across each feature map is taken to reduce spatial dimensions while preserving essential information.

- **Dense Layer with ReLU Activation**:

  A Dense layer with 50 units and Rectified Linear Unit (ReLU) activation is added. This fully connected layer enables the model to learn non-linear combinations of features. ReLu activation function was used to avoid vanishing gradient problems.

- **Dropout Layer**:

  To address overfitting, a Dropout layer with a dropout rate of 0.5 is added. This layer randomly drops out half of the neurons during training, increasing generalizability.

- **Final Dense Layer with Softmax Activation**:

  The last layer is a Dense layer with 12 units, one for each class, and a softmax activation function. This layer produces the final probability distribution over the classes.

2.4 Training algorithm

The optimization algorithm used in the project is the Adam optimizer, an extended variant of SGD. It optimizes the learning rate for each weight in a neural network by utilizing estimations of the first and second moments of the gradient. Adam was chosen over the classic SGD because it is genereally regarded better in terms of generalizing performance and requires less tuning. [2]

The loss used was the 'categorical cross-entropy loss' which is a common choice for multi-class classification. The model's performance during training was evaluated based on accuracy, which is also a common metric for classification tasks.

2.5 Dataset and split

The dataset is composed by 15,150 images. The split among training set, validation set and test set is reported here:

- Training set: 80%
- Validation set: 10%
- Test set: 10%

Other splits were also tried (e.g. 70/10/20) but after a manual inspection of the accuracy the chosen split was proven to be the best for the task at hand.

2.6 Preprocessing

To enhance the model's robustness and prevent overfitting, various image preprocessing and data augmentation techniques have been used through the 'ImageDataGenerator' from Keras. Among the techniques used there were zooming, flipping, rescaling, rotation and many more.

2.7 Training

In each iteration a model is trained using the provided training dataset for a specified number of epochs 15 epochs. During training, the model's performance is monitored on a separate validation dataset. Callbacks, including TensorBoard for visualization, CSV logger for logging training history, early stopping to prevent overfitting and waste less time and model checkpointing to save the best weights, are employed to enhance the training process.

2.8 Hyperparameter search space and selection

To find an appropriate learning rate a learning rate scheduler was initally used but the accuracy at the end of the epochs was not high enough. Thus, an parameterGrid utility was added to iterate over multiple learning rates and find the best one. For each hyperparameter a model was created and compiled. The learning rate of the model with the highest validation accuracy was used for the final model used in testing. It is important to note that the Adam optimizer fine-tunes the learning rate with dynamic adjustments each iteration. The best learning rate found was 2e-4. The mentioned learning rate was used together with a learning rate scheduler callback that halved the learning rate every five epochs, for a faster convergence and more stability, and an early stopping callback with patience 3, for a better performance and avoid overfitting.

Other hyperparameters like the batch size, the number of epochs or the number of units in the dense layer were tuned manually based on the validation accuracy obtained.

3 Results and Discussion

The model's performance was assessed on the test set using the above mentioned learning rate and callbacks, and the obtained results are presented below.

3.1 General results

Overall it can be seen that the model has a great performance across various metrics, demonstrating its effectiveness in accurately classifying garbage images.

Metric	Value
Accuracy	0.9465
Precision	0.9480
Recall	0.9465
AUC	0.9972
F1 Score	0.9465

Table 1: Summary of Model Performance

3.2 Model loss and accuracy by epoch

As expected the model loss decreases and accuracy increases the more the model is trained. Initially, there is a rapid decline in the loss, accompanied by an increase in accuracy. However, as the epochs progress, a plateau effect becomes apparent. This suggests that increasing the number of training epochs may cause only marginal improvements and risk overfitting.

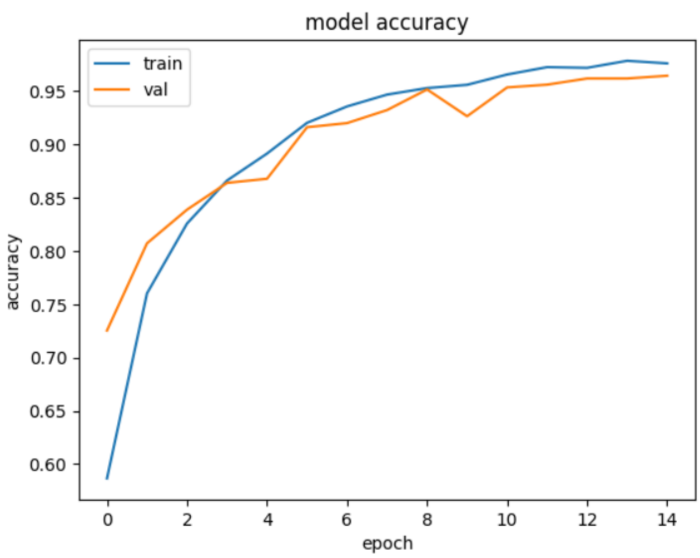

Figure 1: Model accuracy by epoch

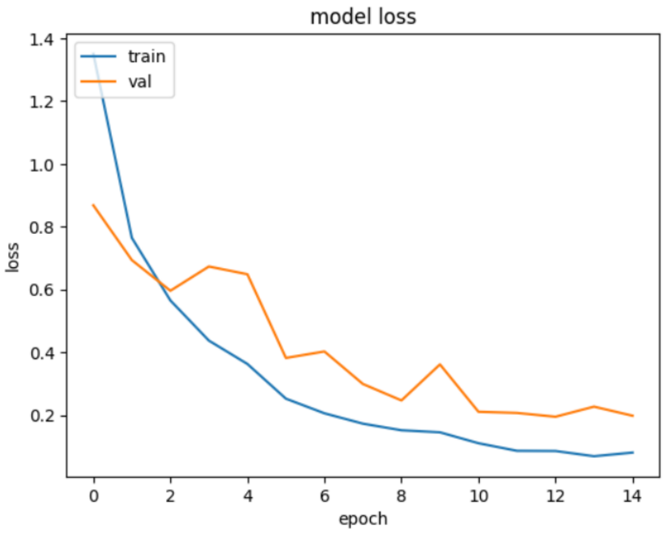

Figure 2: Model loss by epoch

3.3 Class-specific statistics

Here are reported class-specific statistics of the model. As it can be seen from the graphs and table below, even if the dataset is not equally divided among the 12 classes, the model has good performance on every one of them. For example, in the table, we can see that no metric falls below 0.88.

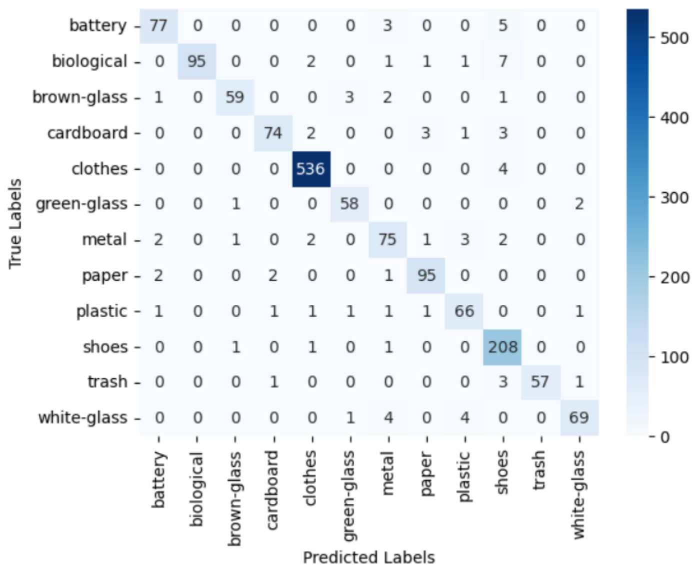

Figure 3: Confusion matrix

Class	Precision	Recall	F1-Score	Support
Battery	0.93	0.91	0.92	85
Biological	1.00	0.89	0.94	107
Brown-Glass	0.95	0.89	0.92	66
Cardboard	0.95	0.89	0.92	83
Clothes	0.99	0.99	0.99	540
Green-Glass	0.92	0.95	0.94	61
Metal	0.85	0.87	0.86	86
Paper	0.94	0.95	0.95	100
Plastic	0.88	0.90	0.89	73
Shoes	0.89	0.99	0.94	211
Trash	1.00	0.92	0.96	62
White-Glass	0.95	0.88	0.91	78
Accuracy			0.95	1552
Macro Avg	0.94	0.92	0.93	1552
Weighted Avg	0.95	0.95	0.95	1552

Table 2: Class-specific metrics

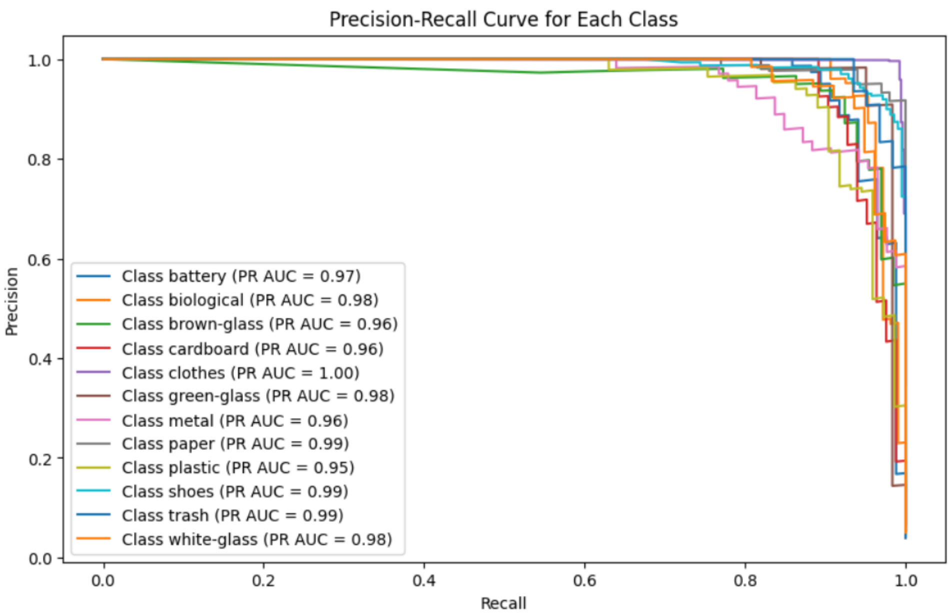

Figure 4: Precision-Recall Curve

3.4 Overall Evaluation

The obtained results suggest that the model, trained on a diverse dataset of household garbage images, performs well in classifying different types of garbage. This demonstrates that the model has a great generalizability and no evident overfitting issue.

4 Conclusion

In conclusion, the automated garbage classification model demonstrated promising results in categorizing household waste.

The great performance proved the model's ability to generalize well across various classes without evident overfitting and it highlighted the effectiveness of the transfer learning paradigm in addressing complex image classification tasks, even with a dataset of limited size.

The outcomes of this project show a real possibility of integration of such automated systems to enhance the efficiency of waste recycling processes, contributing to a more sustainable and environmentally conscious future.

References

[1] Mostafa Mohamed. Garbage classification (12 classes), 2020. URL: `https://www.kaggle.com/datasets/mostafaabla/garbage-classification`.

[2] Akash Ajagekar. Adam, 2021. URL: `https://optimization.cbe.cornell.edu/index.php?title=Adam`.