Waste Classification

By:malak alaa mohamed

id:23012115

Abstracation:

Managing waste properly is one of today's biggest challenges. with the cities growing and consumption increasing the need for better recycling waste handling has been necessary. And technology can help in many ways through automatic waste classification where an AI model identifies the type of waste so it can be recycled correctly.

In this project we used the TrashNet dataset which contains six different classes of waste types:glass,plastic,cardboard,plastic,metal,and trash. We used different machine learning and deep learning models to see which works best for this task we covered six models: CNN classifier,ResNet50 for transfer learning, YOLO for real-time detection, GAN for image generation, Autoencoder for unsupervised feature learning,and Multimodal that combines image and text to improve classification.

Each of these models will be discussed in detail along with their performance,strength,weaknesses,and challenges. Beyond accuracy we will also reflect on the Ethical & Societal Impact of ai waste classification which recycle plants,enable smart bins,and help municipal waste services achieve sustainability goals.

Model 1: Convolutional Neural Network (CNN):

For the first model, I built a simple Convolutional Neural Network (CNN). CNNs are popularly used in computer vision because they automatically pick up on visual features and patterns like edges,shapes,and textures without needing manual feature engineering. This model shows us how well a straightforward deep learning approach performs on waste classification.

How the model was built:

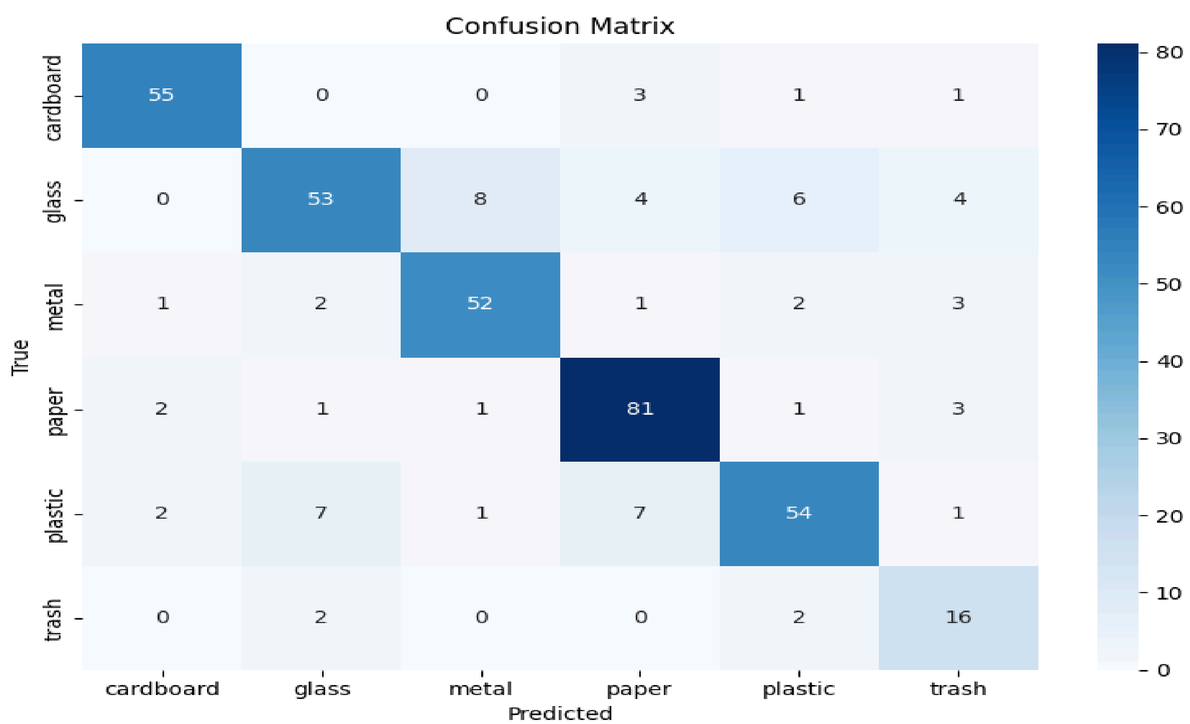
The network was designed with four convolutional layers each followed by batch normalization and max pooling to help the model focus on the important features and patterns in the image.

Next I flattened the output and added a fully connected layer with 256 neurons and a dropout layer to reduce overfitting. For the output layer I added a softmax layer with 6 neurons one for each class (glass,paper,trash,plastic,metal,and cardboard).

I trained the model by using an adam optimizer with a small learning rate and applied data augmentation (rotations,shifts,zooms,and flips) to make the model more robust.
I used early stopping and learning-rate scheduling so the training would stop once the model stops improving.

How the model performed:

The model scored 82% accuracy on the validation data which is strong.
The classification report shows that cardboard and paper were classified very accurately (above 90%).
Plastic and glass were sometimes confused with each other which makes sense because they can look similar sometimes depending on the lighting.
The hardest class was trash which is only around 67% f1-score because trash is mixed class with lots of variety and the class imbalance.



Challenges faced:

- Class imbalance
  Problem: some classes like trash had very few images compared to the other classes which made the model biased towards bigger classes and often misclassified trash.
  Solution: I used class weights during training to give importance to small classes and help the model treat all classes fairly.
- Overfitting:
  Problem: at first the model was performing well on the train set but struggled on the validation set which meant that the model was memorizing instead of learning.
  Solution: I applied data argumentation to make the model more robust. I also added a dropout layer and used early stopping to prevent over training.
- Learning rate tuning:
  Problem: the model sometimes got stuck and accuracy didn't improve.
  Solution: i used a learning rate scheduler called (ReduceLROnPlateau) that automatically lowers the learning rate when validation loss stops improving which allows the model to make better adjustments and reach better accuracy

What i learned:

This model showed that even simple CNN can do a good job at sorting waste but it has its challenges. Certain classes are naturally harder to distinguish and data imbalance makes harder for the model to learn equally.

Model 2: ResNet50(transfer learning):

ResNet50 is a powerful deep learning model that's already trained. Instead of learning from scratch we reused the model's knowledge and adapted it to the Trashnet dataset.
In a process called transfer learning which allows us to build on an already trained model and adapt it to our dataset we utilized resnet50's ability to

recognize edges,texture, and shapes,and make it work with our waste classification project.

I froze the early layers of ResNet50 and only-fine tunes the deeper layers. The reason is that the early layers learn very generic features while the deeper layers learn task specific features by unfreezing those deeper layers the model could learn to distinguish subtle differences like plastic and glass or paper and cardboard.

How the model was built:

I loaded the ResNet50 model with pre-trained ImageNet weights excluding the top classification layers then added my own custom classifier on top:
A global average pooling layer to reduce the output dimensions.A dense layer with 256 neurons and ReLU activation.A dropout layer to prevent overfitting .A final softmax layer with 6 neurons one for each class.
During training i froze the lower layer of
ResNet50 to keep it general image features and only trained the top layers first then i fine tuned more layers at a lower learning rate to adapt the model more specifically to the Trashnet dataset
Then I applied data augmentation (rotations,zooms, flips,and brightness changes). I also applied early stopping to avoid overfitting.

How the model performed:

ResNet50 achieved 89% accuracy on the validation set outperforming CNN baseline.
The classification report showed that:
Glass,cardboard,and paper were predicated very well(above 90% f1-score).
Plastic and metal sometimes got mixed because of different lightning angles.
Trash remained the hardest class with with lower performance than the other classes though better than the CNN.

Challenges faced:

- Training time: ResNet50 is a much deeper so it requires more computing power and time compared to the CNN
- Overfitting risks: at first the model began overfitting quickly due to the smaller dataset size so i had to use dropout, heavy augmentation, and early stopping to control it
- Class imbalance: just like in the CNN the trash class remained difficult because of its variability and fewer examples compared to the other classes

What i learned:

ResNet50 showed the power of transfer learning even without designing a complex model from scratch. I could use a pretrained network and achieve much better results than the baseline CNN. It confirmed that deep architectures capture better details and generalize better but they also come with higher training costs and require careful regularization.

Model 3: YOLOv8 (Object Detection):

Unlike CNN and REseNet50 which only classify an image into one label, yOLOv8 can detect and localize multiple objects within the same image. This makes it very powerful for real world waste sorting where more than one type of item appears together.

How the model was built:

I used Roboflow to download the annotated dataset in YOLO format which provides bounding boxes for each class. Then i trained the YOLOv8s in three stages (chunks of 20 epochs each)

To avoid over heating.

Training included: 60 epoch, batch size 16, image size 640,optimizer: Adam with early stopping for stability, augmentation handled internally by YOLO.

Finally i evaluated the model and tested it on unseen images

How the model performed:

YOLOv8 achieved large detection performance:

Overall results:

- precision=70%
- recall=80%
- mAP50=83.4%
- mAP50-95 =69.7%

Per class results:

- Cardboard: excellent performance (mAP50=96.5)
- paper ,metal,plastic; very strong (mAP50 between 86-91%)
- Glass: decent but weaker (mAP50=80%)
- Trash: lowest performance (mAP50=56.8%-recall only 27%)

This confirms that YOLO is very effective for detecting structured materials but struggles with trash class due to its diversity and unstructured nature.

Challenges faced:

- Hardware limitations: training YOLOv8 pushed my gpu to its limit so i had to split the training chunks and let the device cool down between runs.
- Class imbalance: like ResNet50 and CNN the trash class caused problems because of its inconsistent samples.
- Annotation issues: some bounding boxes in the dataset overlapped or were duplicated which affected the training until cleaned.
- Computation cost: training was slower and required more memory compared to CNN.

What i learned:

YOLOv8 demonstrated how object detection models go beyond classification by actually locating items in the image which is much more practical for automated waste sorting systems.

It also highlights the importance of balance and consistent annotations if one class is too vague or underrepresented like trash even the most advanced models will struggle

Training results:

Model 4: Generative Adversarial Network (GAN):

Gans are not directly used for classification instead their strength lies in creating new synthetic images that can expand and balance the dataset. Since Trashnet has class imbalance like trash i used Gan to generate extra data.

How the model was built:

The Gan consists of two neural networks that compete against each other:
The generator,which takes random noise as input and tries to create realistic looking waste images.
The discriminator, which looks at an image and tries to decide whether it's real from trashnet or fake from the generator.
They are trained together in a back and forth process. The generator gets better at fooling the discriminator while the discriminator gets better at spotting fakes. Over time this training helps the generator produce more realistic images.
I used a standard DCGAN-style architecture with convolutional and deconvolutional layers, batch normalization, and leakyReLU activations.
Training was done for 200 epochs with the adam optimizer, and i periodically saved generated samples to visually track the progress

How the model performed:

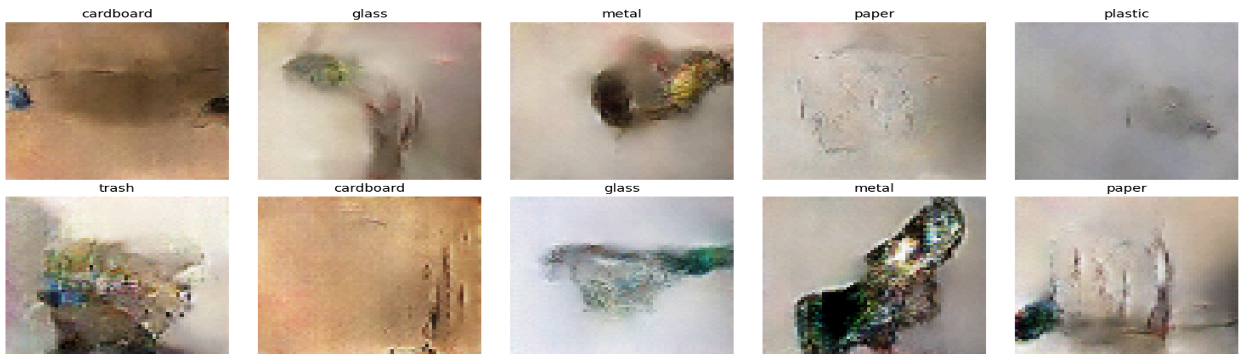
As training advanced the generator gradually learned to produce more coherent and realistic images after 120 epochs the generated output began to visibly resemble waste objects like cardboard and plastic however some classes especially trash were harder for the model to capture accurately.
While GAn don't provide accuracy like CNNs or YOLO their success was judged visually by comparing generated samples to real ones and by noting the diversity and clarity of the synthetic images produced.

Challenges faced:

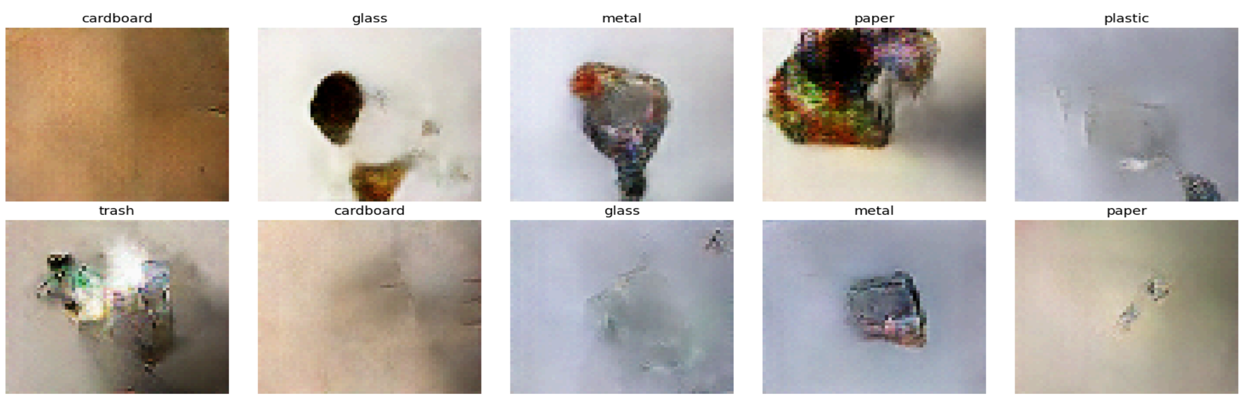
- Training instability: GANs are known for being unstable the model sometimes suffered from mode collapse where the generator produced repetitive outputs

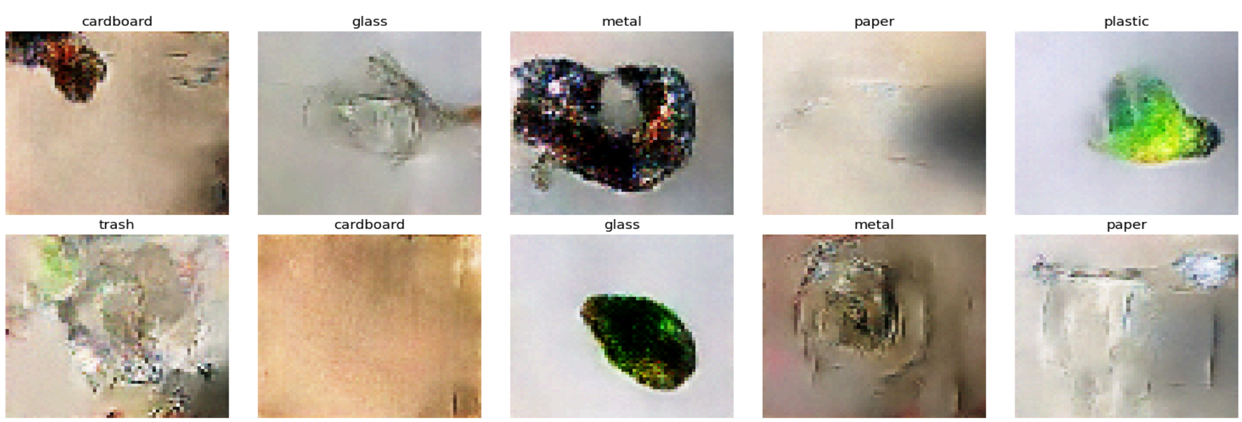
- Image quality: while some images were realistic others contained noise and lacked sharpness, particularly for complex categories like glass and trash
- Evaluation Difficulty: unlike other models there were no clear numeric metric like accuracy of mAo to directly evaluate performance making assessment more subjective
- Resource demand: GAN training required significant gpu time and was slower compared to the other models

Training results



:





Model 5: autoencoder for image denoising:

denoising autoencoder is used to improve the quality of waste classification images by removing noise before feeding them into classifiers. The motivation is that the real world trash images are blurry, poorly lit, or noisy. So by reconstructing cleaner images the downstream classification step could potentially perform better.

How the model was built:

The model was constructed as a custom autoencoder decoder neural network using convolutional and transpose convolutional layers:

Encoder: compressed the input into lower dimensional latent representation
Decoder: reconstructs the denoised image back to the original resolution using transposed convolutions.

Training was performed by adding gaussian noise to the training dataset and asking the model to predict the clean (original) image. The loss function used was mean square error (MSE) and the optimization was done by Adam.

Once trained the autoencoder was used to denoise both training and testing sets. These reconstructed images were then visualized to confirm that the encoder successfully learned to remove much of the added noise.

How It Performed:

The autoencoder was able to effectively denoise images and reconstruct them with clear structure even after adding artificial Gaussian noise. Visual inspection of outputs showed that:

The main shapes and colors of objects were preserved.

Background noise and distortions were reduced significantly.

While some fine details were blurred, the reconstructed images were much cleaner than noisy inputs.

This process demonstrated that autoencoders can serve as a preprocessing step for downstream classifiers such as CNNs and ResNet50.

Challenges faced:

- Training Stability: the autoencoder required careful balancing of learning rate and noise levels. Too much noise degraded reconstruction quality while too little noise limited generalization.

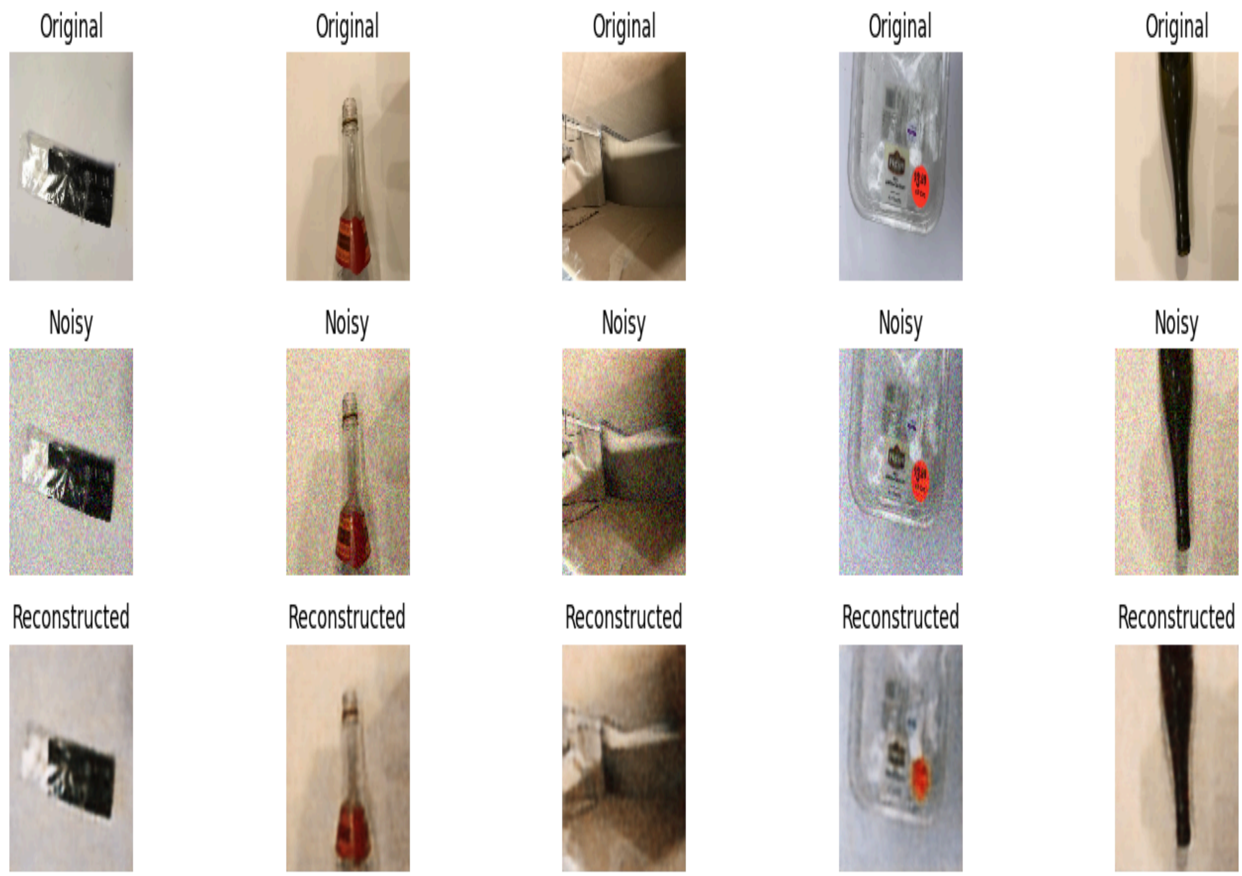
- Blurriness: although noise was reduced the reconstructed images sometimes looked oversmooth using finer details.

- Computational Cost: training the autoencoder added an extra phase before classification increasing total training time.

- Limited Accuracy Boost: while denoising improved input quality it did not always translate into significantly higher classification accuracy when paired with ResNet50.

Despite these challenges the autoencoder provided an important proof of concept that preprocessing through denoising can help make models more robust when handling real world noisy data.

Training results:



Model 6: multimodal fusion (image+text):

This multimodal learning approach combines both image data and textual descriptions to classify waste. The motivation here is that while images give visual cues, additional text-based context can help models better understand and distinguish between confusing categories

How the model was built:

Image Branch: Used ResNet50 pretrained on ImageNet as the feature extractor. Most layers were frozen except the top ones to allow fine tuning. Features were extracted using a Global Average Pooling layer.

Text Branch: Used a TextVectorization layer to tokenize short text descriptions of the waste items. These were passed through an Embedding layer and then pooled with Global Average Pooling 1D.

Fusion: The image and text features were concatenated and passed through a fully connected layer with softmax activation to predict one of the six waste classes.

Loss & Training: The model was trained using Focal Loss to handle class imbalance and optimized with Adam.

How It Performed:

The multimodal approach achieved:

- Validation Accuracy: 91.7%

- Strong performance across all six categories with precision, recall, and F1-scores above 0.85 for most classes.

- The model excelled at differentiating between plastic, paper, and cardboard, where single-modality models sometimes confused them.

This result shows that combining vision and text gives the system a richer understanding of the data leading to improved classification compared to CNN or ResNet50 alone.

Challenges faced:

- Text Augmentation: Since the TrashNet dataset doesn't come with text,I had to generate synthetic textual descriptions. Ensuring variety while keeping descriptions meaningful was key.

- Balancing Modalities: Early experiments showed the model sometimes over relied on text or images alone. Proper normalization and training helped balance both branches.

- Increased Complexity: Multimodal models are heavier and require more memory making them slower to train compared to CNNs or ResNet50.

- Data Quality Mismatch: While images are highly detailed the text was simple. Aligning both types of information in a meaningful way required experimentation.

Despite these challenges the multimodal model achieved the highest overall accuracy (91.7%) proving that integrating multiple data types can significantly enhance waste classification performance.

Ethical and Societal Impact of AI in Waste Classification:

AI in waste classification is not just about building accurate models, it's about making a real difference in how we deal with everyday waste. By using AI, recycling plants can sort materials faster and more safely, reducing the need for people to handle dangerous or unpleasant waste. Smart bins equipped with AI can also guide people to throw things in the right place, helping communities recycle more and waste less.

For cities, this technology has the potential to transform how municipal waste services work. Smarter sorting means fewer trips to landfills, more materials being reused, and less money spent on inefficient processes. In the bigger picture, it supports global efforts to cut carbon emissions and move towards a more sustainable way of living. At its core, AI in waste classification is not just a technical achievement, it's a tool that can help us take better care of our environment and our future.

Real-Life Applications of AI Waste Classification

1. **Recycling Plants**
   - Automates the sorting of materials like plastic, glass, and metal.
   - Reduces human error and speeds up the recycling process.
   - Example: Sorting belts with AI cameras that identify and separate waste.
2. **Smart Bins in Cities**
   - Public bins with built-in cameras can classify waste as recyclable or non-recyclable.
   - Helps citizens dispose of items correctly.
   - Example: Smart recycling stations in airports, malls, or universities.
3. **Municipal Waste Management**
   - AI helps city services track and sort collected waste more efficiently.
   - Reduces landfill usage and improves recycling rates.
4. **Environmental Monitoring & Education**
   - Schools and communities can use AI waste classification apps to teach children about recycling.
   - Mobile apps could scan items and tell users where to throw them.
5. **Industry & Manufacturing**
   - Companies can use AI to sort production waste (e.g., separating metal scraps, plastics).
   - Supports circular economy practices by reusing materials instead of discarding them.