Seyedeh Zahra Alavi, Hamid Sohrabi, Münevver Altinöz

Prof. S. Zigan

PTI51300-Theory to practice und professional skills

August 13, 2024

Westsächsische Hochschule Zwickau

Autonomous street cleaner with integrated waste sorting system

1- Street waste management

Street waste specifically refers to the waste materials that accumulate on streets, sidewalks, and other public spaces. It typically includes a wide variety of debris, much of which is different from household or industrial waste. The composition of street waste can be quite diverse, depending on factors like location, season, and public behavior.

A general overview of street waste composition, particularly relevant to urban environments like those in Germany could be as follows:

1. Organic Litter (30-40%)
   - Leaves and Plant Litter have a high percentage of street litter, especially in the fall, is made up of fallen leaves, grass clippings, and other plant litter.
   - Food Litter: While less common, food litter is still a sizable component in many areas, especially near markets or street food vendors.
2. Plastics (20-25%)
   - Bottles, Packaging Materials, Bags: Plastics are the largest constituent of street litter because of bottles, wrappers, and single-use carry bags.
3. Paper and Cardboard (10-15%)
   - Newspapers, Flyers, Cardboard: Trash paper items like newspapers, flyers, and small cardboard pieces constitute street litter.

4. Glass (5-10%)
    - Broken Bottles, Glass Fragments: Glass is contributed to the waste stream majorly by broken bottles and other glass containers.
5. Metals (5-10%)
    - Aluminum Cans, Metal Scraps: Metal wastes are primarily made up of beverage cans and small metal scraps.
6. Textiles (5-10%)
    - Clothing and Fabric: Dis-carded apparels and fabrics find their way into the street wastes at times.
7. Miscellaneous Waste (5-10%)
    - Cigarette Butts: One of the most ubiquitous and persistent varieties of litter in almost all urban roads.
    - Construction Debris: Small heaps of rubble, sand, and other loose materials from construction activities nearby.
    - Hazardous Waste: At times, it comprises batteries, syringes, and other harmful things.
8. Fine Particulate Matter (5-10%)
    - Dust and Sand: These fine particles accumulate from various sources, including vehicle emissions, construction activities, and naturally occurring dust (Agency, n.d.)

1-1- Main challenges

Sorting street waste after collection presents numerous challenges due to the mixed and contaminated nature of the waste. The diverse materials—ranging from plastics and metals to organic matter and hazardous items—require careful separation, which is complicated by the presence of contaminants like food residues and liquids. Automated sorting technologies often struggle with the irregular shapes and sizes of waste items, leading to inefficiencies, while manual sorting is labor-intensive and costly. Additionally, the handling of hazardous materials poses safety risks, and fluctuations in the market value of recyclables can affect the economic viability of sorting efforts. These challenges highlight the need for advanced technology and improved waste management practices to ensure effective street waste processing.

Traditional street-cleaning methods, often labor-intensive and time-consuming, struggle to keep up with the growing demands of modern cities. Simultaneously, effective waste management requires not only the collection of garbage but also its sorting into different categories for recycling and disposal. Addressing these challenges necessitates innovative solutions that combine advanced technologies in robotics, artificial intelligence, and environmental engineering.

2- Introduction to autonomous street wastes cleaning and sorting machine (ASCWSM)

The concept of an Autonomous Street-Cleaning and Waste-Sorting Machine (ASCWSM), is a cutting-edge device designed to revolutionize urban cleanliness and waste management. The ASCWSM aims to autonomously navigate city streets, clean debris, and classify collected waste into appropriate categories for recycling or disposal. This integration of street-cleaning and waste-sorting functionalities in a single machine presents a significant advancement over existing technologies.

Major components and design considerations of such a device are outlined as follows:

- Robotics platform
- Cleaning mechanism: Sweepers and brushes, vacuum system
- Waste Collection and sorting section consist of conveyor belts, sorting mechanism, compactors
- Sensors and cameras like LiDAR and Ultrasonic Sensors For navigation and obstacle avoidance, high-resolution cameras, computer vision applications include object detection, waste classification, and environmental sensors such as air quality, hazardous material detection, etc.
- Controls System
- Power Supply

A variety of robotic systems have been developed for the purpose of collecting and managing waste from different sources and environments. These include robots that operate in urban areas (MAKESOME, 2020), domestic settings (Volvo Group, 2016), aquatic ecosystems (Mark Rober, 2021), and marine habitats (CNET, 2021). The tasks performed by these robots range from sweeping the streets (Trombia Technologies, 2023) to sorting the garbage (BiN-E | Smart Waste BiN, 2023) according to its material (Robotic Street Cleaner, 2023) but designing a machine to have all reqirements as a autonomous machine to sweep and sort the street wastes needs more studies and advancement specially in the sorting speed and space required for that.

2-1- Latest Research and Developments

Recent studies and developments in robotics and waste management continue to push the boundaries of technology in this field. A notable study published in *Frontiers* showcases an AI-driven robot capable of autonomously detecting and sorting common roadside litter. This system leverages deep learning algorithms to enhance its performance over time, making it highly adaptable to different types of waste materials (Almanzor, Anvo, Thuruthel, & Iida, 2022).

In 2018, the European Union funded the "*HR-Recycler*," project, which aimed to develop a hybrid human-robot recycling plant specifically for electrical and electronic equipment. The objective was to deploy advanced robotic systems across European recycling facilities, enhancing their efficiency and effectiveness (CORDIS, 2024, May 21).

Moreover, a significant study in the *Journal of Material Cycles and Waste Management* delves into the potential of deep learning algorithms to recognize a broader array of waste materials, especially from construction and demolition debris. This research has shown that robots can accurately identify and sort complex materials such as drywall, concrete, and various types of wood, marking a step forward for robotic involvement in managing the diverse waste streams produced by the construction industry (Ku, Yang, Fang, & et al, 2021)

3- Scope of the study

As previously discussed, this advanced technology can be categorized into two primary areas: 1) autonomous driving and waste collection systems, which include sweeping and vacuuming mechanisms on the vehicle, and 2) waste sorting mechanisms and equipment. The focus of our project is on the latter category, specifically addressing the development and optimization of waste sorting technologies.

Interest in Artificial Intelligence, Machine Learning, and Deep Learning is surging in today's world. These fields employ a variety of sophisticated techniques and algorithms to achieve remarkable outcomes. Deep learning, in particular, utilizes neural networks to replicate the functioning of the human brain. These neural networks are applied across a wide range of deep learning applications, including image processing, image segmentation, and autonomous vehicles. One of the most prominent types of neural networks is the Convolutional Neural Network (CNN), known for its effectiveness in these areas.

In our project, we will leverage the CNN method to enhance the capabilities of the ASCWSM. CNNs are particularly well-suited for image classification tasks, which is essential for accurately sorting different types of waste. Throughout this report, we will clearly explain every step of our machine learning process, detailing how CNNs are applied to achieve efficient and reliable waste categorization. This approach not only ensures a high level of accuracy in waste sorting but also demonstrates the potential of integrating advanced AI techniques into practical environmental solutions.

The codes of Python used in this project can be accessed from this link:

https://colab.research.google.com/drive/15dPGo9BpdIR4xf0fn7nTRsTRoUEN4W0n?usp=sharing

4- An introduction to Convolutional Neural Networks

Convolutional neural networks (CNNs) derive their name from the mathematical operation of convolution, which involves combining two functions to produce a third function. In CNNs, convolutional layers apply filters or kernels to the input data to create feature maps that capture various patterns or aspects of the data. CNNs are composed of multiple layers arranged in sequence. The basic structure includes an input layer, several convolutional layers, pooling layers, a fully connected layer, and an output layer. The convolutional layers are key to extracting features from the input data by applying filters. Pooling layers reduce the spatial dimensions of these feature maps, thereby decreasing computational complexity. The fully connected layer links the extracted features to the output layer, enabling the network to make predictions or classifications based on the learned features (Ahad, Li , Song, & Bhuiyan, 2023).

The input and output layers are known as the visible layers of a CNN, while the intermediate layers, such as convolutional and pooling layers, are called hidden layers. These hidden layers are crucial for learning and extracting hierarchical representations from the input data. The combination of convolutional, pooling, and fully connected layers in CNNs allows the network to effectively learn and identify patterns in complex data, making CNNs particularly well-suited for tasks like image recognition and computer vision (Xie & et al, 2017).

4-1- Libraries and Their Purposes

For building and training a CNN using TensorFlow and Keras, we imported essential libraries. We used NumPy for handling large, multidimensional arrays and numerical operations, while Pandas was utilized for data manipulation and analysis. Matplotlib.pyplot helped in visualizing data distributions and model performance. TensorFlow and its high-level API Keras were crucial for constructing and training the neural network models, with specific utilities like Sequential for stacking layers and layers such as Conv2D, MaxPool2D, Flatten, Dense, and ZeroPadding2D to build and refine the CNN. Additionally, we employed

ImageDataGenerator for data augmentation and callbacks like EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint to monitor and optimize the training process.

4-2- Dataset Exploration and Understanding

After importing the necessary libraries and loading the dataset, we proceeded to explore and understand the dataset's structure and content. First, we listed the contents of the data directory, revealing six categories: paper, plastic, metal, trash, glass, and cardboard. We then counted the number of files in each category to assess the distribution: paper contained 594 files, plastic 482, metal 410, trash 137, glass 501, and cardboard 403. Additionally, we calculated the total number of files across all subfolders, finding a total of 2,527 files. This step provided a comprehensive overview of our dataset's size and distribution.

To gain a visual understanding of the dataset, we displayed sample images from each category. This allowed us to confirm that the images were correctly categorized and to inspect their quality. By examining these samples, we ensured that our dataset was appropriately structured and ready for the next stages of model training and evaluation. This preliminary exploration was crucial for identifying any potential issues and preparing our data for effective training of the CNN model.

4-3- Setting Parameters for Model Training

We then defined key variables required for model training. These include the batch size, target size for the images, and the validation split. These parameters were set to ensure that the model would process the data efficiently and effectively. The batch size of 30 was chosen to balance memory usage and training speed. The target size of 180x180 pixels was selected to standardize the input size for the CNN, ensuring consistency and optimal performance. Finally, a validation split of 15% was determined to allocate a portion of the data for validation purposes, which is essential for monitoring the model's performance and preventing overfitting during training.

4-4- Data Augmentation and Generation

To further enhance our dataset and improve the model's robustness, we utilized data augmentation techniques through data generators. Data augmentation helps in creating variations of the existing images, which aids in making the model more generalized and less prone to overfitting. For training data, we applied several augmentation techniques such as rescaling, flipping, zooming, shifting, rotating, and shearing. These augmentations help in creating diverse training samples by randomly transforming the images, which can significantly improve the model's ability to generalize to new, unseen data.

For the validation data, we only applied rescaling to normalize the pixel values. This ensures that the validation data remains realistic while still being normalized for consistent input to the model.

4-5- Split Train & Test Files

To effectively train and validate our model, we divided our dataset into training and validation subsets. We used image data generators to manage this process, which also applied various transformations to the images to make the model more robust. For the training data, we set up a generator that loaded images from the dataset, applied augmentations like flipping, zooming, shifting, rotating, and shearing, and processed them in batches. This generator found and loaded 2,150 images, which were categorized into six different classes. For the validation data, we set up a similar generator but without the augmentations. This ensured that the validation images remained realistic and consistent for evaluating the model's performance. The validation generator found and loaded 377 images, also categorized into six classes.

By splitting the dataset this way, we ensured that our model would be trained on a varied and augmented set of images, while being validated on a realistic subset, helping us monitor performance and prevent overfitting.

4-6- Network Architecture

To create an effective convolutional neural network (CNN) for our garbage detection task, we designed a model with multiple layers that progressively extract and learn features from the images. We started with an input layer that takes images of size 180x180 pixels with 3 color channels (RGB) then a ZeroPadding2D layer was added to pad the borders of the input with zeros, which helps in maintaining the spatial dimensions after convolution operations. We added three convolutional layers, each followed by a max-pooling layer and a dropout layer:

- The first convolutional layer has 32 filters, each of size 3x3, and uses the ReLU activation function. It is followed by a 2x2 max-pooling layer and a dropout layer with a 30% dropout rate.
- The second convolutional layer has 64 filters, each of size 3x3, with ReLU activation. This is also followed by a 2x2 max-pooling layer and a 30% dropout layer.

- The third convolutional layer has 128 filters, each of size 3x3, with ReLU activation. This is followed by a 2x2 max-pooling layer and a 40% dropout layer.

After the convolutional layers, a Flatten layer was used to convert the 2D feature maps into a 1D feature vector. We added two fully connected (dense) layers:

- The first dense layer has 128 units with ReLU activation, followed by a dropout layer with a 50% dropout rate.
- The second dense layer has 64 units with ReLU activation.

Finally, a dense output layer with 6 units (corresponding to the 6 classes of garbage) and a SoftMax activation function was added to produce the class probabilities.

This architecture ensures that the model can learn complex patterns and features from the input images while regularization techniques like dropout help in preventing overfitting.

We also visualized the model architecture (Figure 1) to confirm the structure and layer configurations:

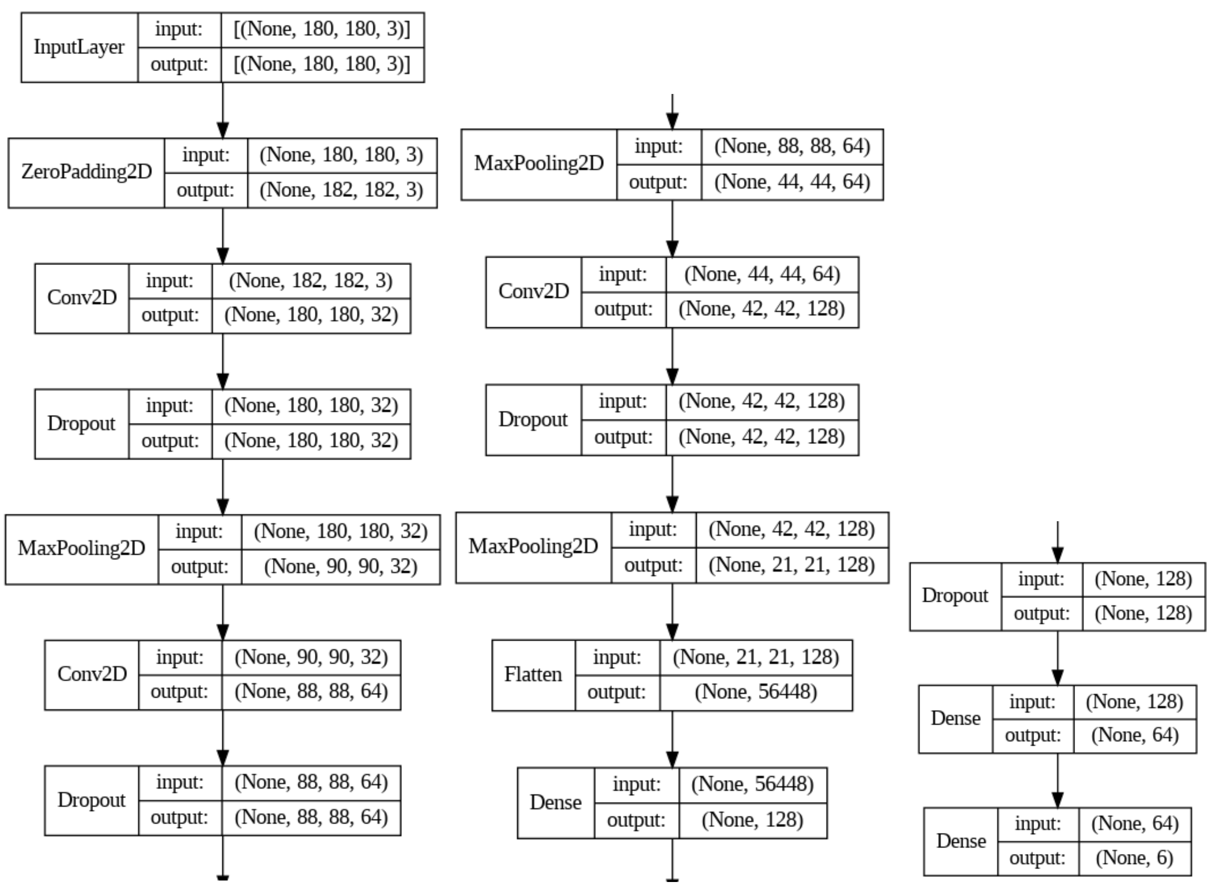

Figure 1-model architectur

4-7- Model Compilation

To prepare our model for training, we implemented several key configurations. We used the Early Stopping callback to monitor the validation loss and stop training if it didn't improve for 10 consecutive epochs, which helps prevent overfitting and saves time. The Reduce LROnPlateau callback was added to automatically reduce the learning rate by a factor of 0.1 if the validation loss stopped improving for 5 epochs, ensuring better learning efficiency. Additionally, we set up the Model Check point callback to save the best version of the model whenever there was an improvement in validation loss, storing it in a file named 'trash.h5'. Finally, we compiled the model with the 'adam' optimizer for efficient training, the Categorical Cross entropy loss function suited for our multi-class classification task, and Categorical Accuracy as the metric to monitor performance throughout training. These configurations ensure that our model is well-optimized and capable of generalizing effectively to new data.

4-8- Train Model

To train our model, we used the model.fit function, which starts the training process using the previously defined settings and configurations. Here's a summary of what we did:

Training Data: We used the train_data generated from our training dataset.

- **Steps per Epoch**: We set the steps_per_epoch to the length of train_data, which ensures that each epoch processes all training images.
- **Epochs**: We trained the model for 80 epochs, providing ample opportunity for the model to learn from the data.
- **Validation Data**: We used the test_data generated from our validation dataset to evaluate the model's performance during training.
- **Validation Steps**: We set the validation_steps to the length of test_data, ensuring that each validation round processes all validation images.
- **Callbacks**: We included the previously defined callbacks (EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint) to optimize the training process and save the best model.

The training process involved feeding the model with the training data, adjusting the model's parameters to minimize the loss, and periodically evaluating the model's performance on the validation data. The use of callbacks ensured that the training process was efficient and that the best model was saved.

By the end of the training, the model had learned to classify the images into the six categories with an optimized performance, monitored and guided by our carefully set up callbacks.

5- Model Evaluation

After training the model, we evaluated its performance by identifying the highest validation accuracy achieved, which indicated the model's best performance on unseen data. The best validation score was 0.6790, or approximately 67.9%. We then examined the accuracy trends over all epochs for both training and validation datasets. By plotting these accuracy trends, we could visually assess the model's learning progress, observe improvements over time, and check for any signs of overfitting. This comprehensive evaluation provided valuable insights into how well the model learned to classify the images and its generalization capabilities. In Figure 2, you can see the training and validation accuracy over the epochs, which helps in understanding the model's performance throughout the training process.

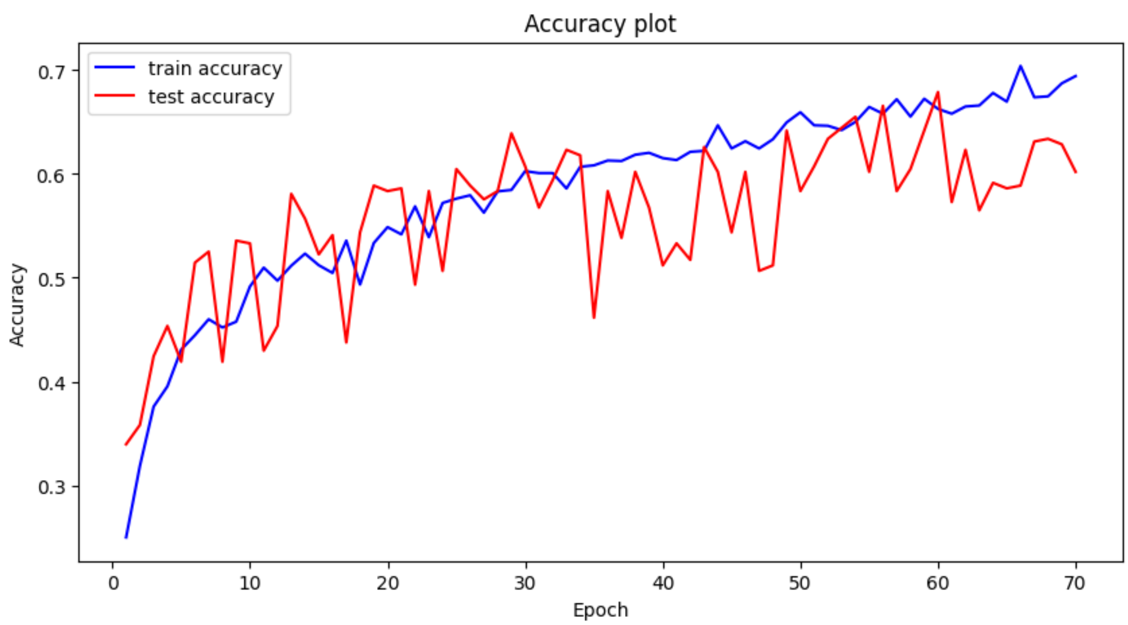

Figure 2- Training and Validation Accuracy Over Epochs

6- Model utilization

To utilize the trained model for classifying new images, we followed a series of steps. First, we obtained the class indices from the training data, which provided a mapping of class names to numerical labels. The output mapping was as follows: {'cardboard': 0, 'glass': 1, 'metal': 2, 'paper': 3, 'plastic': 4, 'trash': 5}. Next, we loaded an image of a plastic bottle using the utils.load_img function. This function ensured the image was in RGB format and resized it to the target size of 180x180 pixels. After loading, the image was displayed for verification. We then converted the image to an array format using utils.img_to_array and normalized the pixel values by dividing by 255.0. This normalization step was crucial for maintaining consistency with the model's training process. To make a prediction, we used the model's predict function on the processed image. The model's output was a set of probabilities for each class, and we determined the class with the highest probability using np.argmax. Finally, by mapping the predicted class index back to the class name using the classes dictionary, the model successfully identified the image as "plastic." The output confirmed this prediction (Figure 3):

```
img = utils.load_img(
    '/content/plastic_072.jpg',
    grayscale=False,
    color_mode='rgb',
    target_size=(180, 180),
    interpolation='nearest',
    keep_aspect_ratio=False
)

img
```

```
img = utils.img_to_array(img)
img = img / 255.0    # Normalize the image

result = model.predict(tf.expand_dims(img, axis=0))

classes = train_data.class_indices

for key, value in classes.items():
    if value == np.argmax(result):
        print(key)
```

```
1/1 [==============================] - 0s 319ms/step
plastic
```

Figure 3- Plastic identification

7- Conclusion

Urban environments worldwide face significant challenges in maintaining cleanliness and managing waste efficiently. Traditional methods are labor-intensive and struggle to meet modern demands. Effective waste management requires not only garbage collection but also its segregation for recycling and disposal. Innovative solutions that combine advanced technologies in robotics, artificial intelligence, and green engineering are essential.

This report introduces the Autonomous Street-Cleaning and Waste-Sorting Machine (ASCWSM), designed to revolutionize urban cleanliness and waste management. The ASCWSM autonomously navigates streets, cleans debris, and classifies waste into appropriate categories for recycling or disposal, presenting a significant advancement over existing technologies.

Interest in AI, Machine Learning, and Deep Learning is growing rapidly. These fields use sophisticated techniques and algorithms to achieve remarkable outcomes. Deep learning, particularly Convolutional Neural Networks (CNNs), is highly effective for image classification tasks. Our project leverages CNNs to enhance ASCWSM's capabilities, ensuring accurate waste sorting.

Throughout this report, we have detailed every step of our machine learning process, demonstrating the efficiency and reliability of CNNs in waste categorization. This integration of advanced AI techniques into practical environmental solutions not only improves accuracy in waste sorting but also showcases the potential for smarter, cleaner cities.

The ASCWSM offers a comprehensive solution to urban waste management challenges, reducing labor and time required for street cleaning and improving recycling efficiency. This technology underscores the importance and feasibility of integrating AI and machine learning into green engineering for a cleaner future.

8- References

1. Agency, G. E. (n.d.). https://www.umweltbundesamt.de/publikationen.

2. Ahad, M. T., Li , Y., Song, B., & Bhuiyan, T. (2023). Comparison of CNN-based deep learning architectures for rice diseases classification. Artificial Intelligence in Agriculture, Volume 9, 22-35.

3. Almanzor, E., Anvo, N. R., Thuruthel, T. G., & Iida, F. (2022). Autonomous detection and sorting of litter using deep learning and soft robotic grippers. Front. Robot. AI, Computational Intelligence in Robotics.

4. CORDIS. (2024, May 21). Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment | HR-Recycler Project | Fact Sheet | H2020 | CORDIS | | European. European Commission. .

5. Ku, Y., Yang, J., Fang, H., & et al. (2021). Deep learning of grasping detection for a robot used in sorting construction and demolition waste. Mater Cycles Waste Manag 23, 84–95.

6. Xie, S., & et al. (2017). Aggregated residual transformations for deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (pp. 1492–1500).