applied sciences

MDPI

Article

Target State Classification by Attention-Based Branch Expansion Network

Yue Zhang [1,2,3], Shengli Sun [1,3,*], Huikai Liu [1,2,3], Linjian Lei [1,2,3,4], Gaorui Liu [1,3] and Dehui Lu [1,3]

1 Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China; zhangyue8@mail.sitp.ac.cn (Y.Z.); liuhuikai@mail.sitp.ac.cn (H.L.); leilj@shanghaitech.edu.cn (L.L.); liugaorui@sitp.ac.cn (G.L.); datcldh@mail.sitp.ac.cn (D.L.)
2 School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
3 Key Laboratory of Intelligent Infrared Perception, Chinese Academy of Sciences, Shanghai 200083, China
4 School of Information Science and Technology, ShanghaiTech University, Shanghai 200083, China
* Correspondence: palm_sun@mail.sitp.ac.cn

Abstract: The intelligent laboratory is an important carrier for the development of the manufacturing industry. In order to meet the technical state requirements of the laboratory and control the particle redundancy, the wearing state of personnel and the technical state of objects are very important observation indicators in the digital laboratory. We collect human and object state datasets, which present the state classification challenge of the staff and experimental tools. Humans and objects are especially important for scene understanding, especially those existing in scenarios that have an impact on the current task. Based on the characteristics of the above datasets—small inter-class distance and large intra-class distance—an attention-based branch expansion network (ABE) is proposed to distinguish confounding features. In order to achieve the best recognition effect by considering the network's depth and width, we firstly carry out a multi-dimensional reorganization of the existing network structure to explore the influence of depth and width on feature expression by comparing four networks with different depths and widths. We apply channel and spatial attention to refine the features extracted by the four networks, which learn "what" and "where", respectively, to focus. We find the best results lie in the parallel residual connection of the dual attention applied in stacked block mode. We conduct extensive ablation analysis, gain consistent improvements in classification performance on various datasets, demonstrate the effectiveness of the dual-attention-based branch expansion network, and show a wide range of applicability. It achieves comparable performance with the state of the art (SOTA) on the common dataset Trashnet, with an accuracy of 94.53%.

Keywords: technical state requirements; target state classification; branch expansion; dual-attention module; parallel residual connection; stacked block

check for updates

Citation: Zhang, Y.; Sun, S.; Liu, H.; Lei, L.; Liu, G.; Lu, D. Target State Classification by Attention-Based Branch Expansion Network. *Appl. Sci.* **2021**, *11*, 10208. https://doi.org/10.3390/app112110208

Academic Editor: Antonio Fernández

Received: 26 September 2021
Accepted: 29 October 2021
Published: 31 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Particle redundancy is not strange in the industrial field; industrial workshops and intelligent manufacturing plants are facing the problem of particle redundancy. However, there are almost no computer vision (CV) methods to detect and control it, due to its randomness and suddenness. At present, the method commonly used in the industry is particle impact noise detection [1,2] at the end of the assembly to identify the existence of loose particles. We tried to track the detection target state through the CV method to find the generation of particle redundancy. We define it in the CV field through the on-site visit as "objects that do not meet the current task technical state requirements". In this article, we pay attention to the tools' technical state and the staff's wearing state and try to provide a solution of redundancy control from the perspective of monitoring management.

Due to fewer application scenarios, object state classification is rarely considered and explored. A few efforts focused on the actions and time sequence that cause the object state transitions [3,4]. Some studies describe objects by their attributes and make fine-grained distinctions across object categories [5–7]. He, Wang et al. [8] proposed a state-tracking detection pipeline for 3D reconstruction of video sequences. They used Faster R-CNN to detect the objects in each frame and used SVM (support vector machines) to classify the contents in the bounding boxes. The states we describe in this article are somewhat similar to the properties in [7,9] (e.g., a whole apple and a peeled apple), but are closer to the external states of an object (e.g., a bunch of scattered screws and fixed screws). Our work is to classify the wear state of the staff and the technical state of the experimental tools. We first collect two datasets: wearing-state and objects-state corresponding to the staff's wearing state and tools' existing state introduced in Section 3.1. Some samples are shown in Figure 1 from which the challenge of the datasets can be discovered: the intra-class distance of the same category is large, but the inter-class distance of different categories is small, which makes it difficult to exclude confounding features. Additionally, the key features as the classification criteria are difficult to distinguish. A proper classification network is required. We designed the datasets based on how Trashnet [10] is organized since it is a relatively small dataset for trash classification and can be used in foreign object debris (FOD) detection, which shares commonalities with particle redundancy detection and target classification in our task.

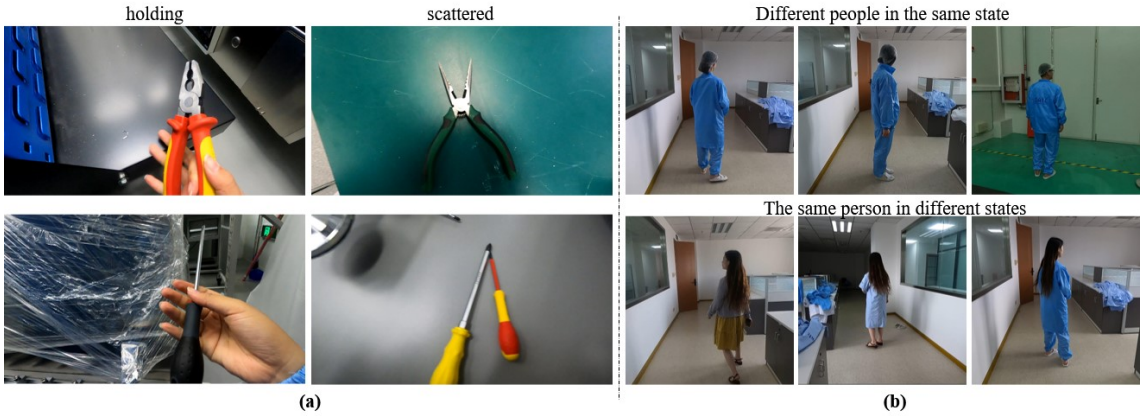


Figure 1. Examples of the two datasets. The left sub-graph (**a**) is from "object-state" showing different tools for the same state (column) and the same tool for different states (row). The right sub-graph (**b**) is samples of different people in the same state (1st row) and a person in different states (2nd row) from the "wearing-state".

In past decades, deep learning accessed a period of rapid development and has surpassed many traditional CV methods–based approaches in various tasks, such as action recognition, semantic segmentation, object detection, etc., due to its tremendous capabilities in learning expressive representations of complex data [11]. Deepening the network can improve the model of linear expression ability, which learns more complex transformation, and better fit the characteristics of more complicated input [12]. While a network with enough width is beneficial for distributed computation [13] and can improve the utilization of each layer [14] so that it learns abundant features, such as texture, colors, etc. Common networks choose to deepen the depth of the network because to a certain extent, deepening the network usually achieves higher performance benefits [15]. However, deeper is not always better; excessively deep networks need stronger computing power and more train time, and even a degradation problem maybe occur [16]. Some works have shown that shallow and wide networks with more channels may work better than deep and narrow ones, or at least as much performance [13,17]. Thereofre, choosing the depth and width of the network to achieve the best effect is a problem worth considering. Our datasets are low in the distinction between feature information; therefore, we try to explore

whether considering the depth and width of the network can improve the classification accuracy. We propose a novel attention-based branch expansion classification network with modified Xception as the backbone. The Xception architecture [18] is a linear stack of depthwise separable convolution layers, which makes the architecture very easy to define and modify. We modify and expand the middle structure of Xception and compare the network performance of different widths and depths to find the backbone structure that best represents the characteristics of our datasets. In addition, which backbone best fits the attention module applied in stacked block mode is explored.

In addition, as can be seen from the samples in Figure 1, different states of the same object belong to different classes, and the different wear situations of the same person are different categories. Considering the particularity of the task of state classification and the challenge of the above-mentioned datasets, we need to explore more effective classification methods to extract activation features and suppress interference features. Therefore, we attach an attention mechanism to distinguish feature information effectively, strengthen key features and suppress confounding features. The attention mechanism used in computer vision is generally divided into spatial attention and channel attention. The former learns the appearance and location information and decides "where" to focus in the image, while the latter focuses on "what" is meaningful in a given image. Inspired by [19], we adopt the plug-and-play dual-attention module with negligible parameter overhead. The difference is that we modify the fusion of the two attention maps with a parallel residual connection, which can achieve a better effect on our datasets.

The objective of this article is to solve the problem of the state classification of objects. Section 2 presents the related work of three categories of image classification. Section 3 describes the proposed method, and the appropriate depth and width of network structure. The application mode of the attention module is explored. Section 4 makes an extensive comparison between the eight models described in Section 3 and selects the most appropriate model structure. Section 5 discusses the correspondence between the experimental results and visualization conclusion, as well as the comparability with the SOTA methods. Section 6 summarizes the work of the article, reveals the shortcomings of the model, and gives possible future solutions. The contributions of our work are as follows:

- We introduce two datasets, "wearing-state" and "object-state", the former with different staffs wearing different lab suits and the latter consisting of different states of different categories tools. The datasets are collected for technical state detection in the laboratory.
- Three structures of Xception-based feature extraction backbones, with different stacked blocks, each consisting of parallel branches, are proposed to better express complex information. We propose four ABE networks based on the four backbones (including Xception) by adopting a plug-and-play dual-attention module with the parallel residual connection.
- We conduct extensive ablation analysis on three datasets to explore the influence of depth and width on the classification performance, and prove the effectiveness of width broadening and a dual-attention mechanism through quantitative comparison and qualitative visualization. Our experiments show consistent improvements in classification performance on various datasets.

2. Related Work

Related methods of state detection include modeling manipulation action through state transformation [3,4,20], object attributes description [5,6,9], image classification [8], etc. Among them, state transfer only focuses on the action causing the state change rather than what the specific state is. Object attribute description tends to describe the given appearance of a class of objects. Closer to scene understanding in [8], our practical application requires a classification of tools' existence state and the staff's wearing state in the scene, which is a multi-object and multi-state classification task. Below, we review the relevant methods of

image classification tasks. Comparisons are given in Table 1, where the data for the last column are explained in detail in Section 5.2.

Table 1. Comparison of related methods.

Method	Year	Performance	Accuracy on TrashNet
KNN	1967	Simple and efficient still used today	88 [21]
SVM	1995	Represents a victory for Kernel technology	80 [21]
RF	2001	The concept of integrated learning	85 [21]
AlexNet	2012	Champion in 2012 ILSVRC	91 [21]
GoogleNet	2014	Champion in 2014 ILSVRC	87.71 [22]
VGG-16	2014	Runner-up in 2014 ILSVRC	93 (fine-tuned) [21]
ResNet	2015	Champion in 2015 ILSVRC	88.66 [22]
DenseNet	2017	CVPR's Best Papers of 2017	89 [23]
Xception	2017	Great progress in GoogleNet series	91.41

2.1. Traditional Methods

The traditional methods of image classification typically include four steps: image preprocessing, feature description, training classifier, and classification identification. Image preprocessing includes normalization, resize and noise elimination, etc., which enhance image quality to avoid unnecessary noise interference and make the model better learn the effective feature. In the late 1970s, Vapnik proposed the support vector method to resolve the pattern identification problem, which then became widely used in machine learning applications, especially image classification [24–26]. SVM is a binary classification model which is a linear classifier defined in the feature space with the largest interval. The kernel-based theoretical system of SVM makes it actually a nonlinear classifier that can be customized [27]. Breiman et al. developed an ensemble learning approach, random forest (RF), to solve classification and regression problems [28], which is robust for missing data and unbalanced data. K-nearest neighbor (KNN) is a simple implementation with significant classification performance [29] because it is nonparametric without a training stage. In fact, traditional machine learning methods, such as SVM and RF, have a better solution on a small amount of training data [28,30]. However, they usually require the manual design of characteristics with complex multi-step operations.

2.2. Deep Learning Methods

The philosophy behind deep learning is to mimic the way a child perceives the world from birth, learning to recognize objects as he/she grows up and processes large amounts of data [31]. An intuitive and convenient way is to use the advanced object detection network for image classification prediction, such as classical two-stage model Faster-RCNN and end-to-end model YOLOv4 [32]. However, fine-grained object-level annotations are required for state classification tasks. The application of deep learning in image classification tasks mainly includes the following models.

In 2012, AlexNet [33] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC); it has five convolutional layers followed by three fully connected (FC) layers with a dropout layer after each FC layer to reduce overfitting. This was also the first time that deep learning was used for large-scale image classification. After AlexNet, a series of CNN models emerged. VGG [34] was developed by the Visual Geometry Group of Oxford in 2014. It has two structures: VGG16 and VGG19 with 16 and 19 hidden layers, respectively. Compared with AlexNet, VGG uses several continuous stacked 3×3 convolution kernels to replace the large convolution kernel in AlexNet (11×11, 7×7, 5×5), because multiple nonlinear layers can increase the depth of the network to ensure learning more complex features. GoogleNet [35] is the 2014 ILSVRC champion model; a modular structure named Inception was proposed to maintain the sparsity of the neural network structure and make full use of the high computational performance of the dense matrix by clustering sparse matrix into more dense sub-matrices. Another special design is

to use the global average pooling (GAP) layer instead of the FC layer to reduce the number of parameters. However, with the deepening of the network, the gradient disappearance leads to degradation of network performance. The Resnet [16] proposes a residual structure that uses a shortcut connection to connect the input and output, which learns the residual to solve the degradation problem. ResNet won five champions at the ILSVRC & COCO 2015 Competitions. CNN showed performance improvements over SVM and RF [36], due to its ability to retrieve complex features and information from the input images on large sample datasets [30].

A bad rule of thumb is that the deeper the network, the better the result, but channel width is important for certain recognition tasks [37]. The authors in [18] used Xception as the backbone. Xception is another structure proposed by GoogleNet based on Inception V3, decoupling channel correlation and spatial correlation, and deriving depthwise separable convolution to replace the original convolutional operation in Inception V3, which simplifies the model and improves the performance. Its architecture is very easy to define and modify, which meets the design requirements of our tasks. We use the core structure of Xception as the basis for a compromise between depth and width.

2.3. Attention

Attention mechanisms have received widespread attention in the field of computer vision research, as they allow models to dynamically focus on related parts of the input. There are two commonly used attention mechanisms in the convolutional neural network. One is spatial attention, which outputs a $C(channel) \times H(height) \times W(weight)$ plane feature map for each layer of the convolutional neural network. The other is channel attention, which learns different weights for each channel. Some works apply mixed attention of the two that focuses on both spatial features and channel features [19,38–40]. Wang et al. [38] proposed a residual attention network for image classification with the structure of stacked multiple mixed attention modules as well as the residual learning mechanism to train the very deep network. The bottleneck attention module (BAM) [39] and convolutional block attention module (CBAM) [19] were proposed by the same team for image classification as a plug-and-play module. Both of them use a mixed attention mechanism; the former connects spatial attention and channel attention in a parallel manner, while the latter connects them sequentially. Generally speaking, spatial attention and channel attention learn what and where to emphasize, respectively, and refine intermediate features effectively. We try to discover the ability of the attention machine to express confounding features in our datasets.

3. Proposed Methods

3.1. Datasets

In order to mimic the monitoring of intelligent robots on laboratory technical states, we collected two new datasets from our digital assembly test center (DTAC) in the manner of fly-around video [8], which allows the camera, around a single class of object, to obtain data at different angles, with each video containing only one state of the object. We downsampled the video frames to remove adjacent frames that are too similar. To ensure that the state recognition is a computer vision problem, we removed irrelevant data, such as creation dates, and we collected sample data from different backgrounds and different lighting conditions. We designed the two datasets in the same way that the Trashnet dataset [10] is composed. The state classification process takes images of a single state class of a single type of object or person and classifies them to the correct category. A comparison between the two datasets and the common dataset, Trashnet, is given in Table 2. The total number of images, number of classes, number of images per class, number of objects, and number of states are compared.

Table 2. Comparison of our proposed two datasets and one common dataset, Trashnet.

Datasets	Total	Class	Each Class	Object	State
Trashnet	2527	6	400–500	6	/
Object-state	6984	14	300–700	5	4
wearing-state	5673	8	500–800	12	8

"Object-state" consists of the different presence states of different objects. There are 6984 images in total belonging to 14 classes of state, with about 300–700 images each class across 5 objects, which is unbalanced. The 5 classes of objects are "screw", "screwdriver", "pliers", "wrench" and "little wrench". The states include "held", "left", "store away" and "fixed". The train and test datasets are split on the type of tool or the background to ensure that the two sets repel each other. The collection process is performed in different working positions under different lighting conditions to introduce variation to the dataset. The sub-graph on the left of Figure 1 shows four sample classes of "object-states", including different states for the same object (each row) and the same state for different objects (each column). It can be found that the same tool may be in different classes, resulting in a small inter-class gap, while different models, colors, backgrounds, etc., lead to a large intra-class gap for the same class.

Another is the "wearing-state" that records whether a worker is wearing their work clothes correctly, which contains 5673 images in total belonging to eight classes with about 500–800 images in each across 12 people. We shoot the wearing state of the workers in different spatial backgrounds with different lighting conditions in a similar way to the former dataset. The train and test data are split on the staff such that one person can only appear on either the train or test set. The sub-graph on the right of Figure 1 shows samples of "wearing-state", including different people in the same state (1st row) and the same person in different states (2nd row), which reflects the challenge of this dataset, i.e., large intra-class difference and small inter-class difference. We propose an attention-based branch extension network for the above two datasets.

3.2. Methods

3.2.1. Branch Expansion Modules

In this section, we build 3 new CNN structures with different depths and widths based on Xception [18], whose core structure is designed to be easily defined and modified to explore how networks of different depths and widths perform differently on our datasets.

The middle flow of the Xception structure is shown in Figure 2, including entry flow, middle flow, and exit flow. The middle flow is a linear stack of a 9-layer depthwise core structure, which has 3 separable convolution layers each after a Relu layer, followed by a batch normalization layer. In order to explore the influence of depth and width on the model, we only modify the repetitive structure "Middle Flow" and maintain the "Entry Flow" and "Exit Flow" unchanged. We extract the 9-layer core structure as an active component, represented in Figure 2 as a colored block, to be reorganized into different structures.

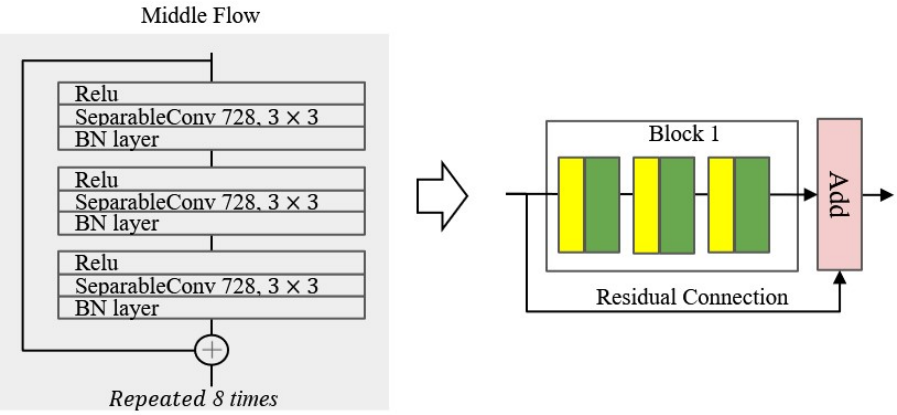


Figure 2. Extract the middle flow of the Xception structure. Note that the middle flow is a linear stacked structure, repeated eight times, extracted and represented as colored parts for new modifications and reorganizations below.

To explore the influence of depth and width on the performance of convolutional neural networks, we reorganize the core structure of Xception in three additional ways. The 4 structures, including Xception itself, are shown in Figure 3 in decreasing order of depth. The "Entry Flow" and "Exit Flow" here are folded up, and we only discuss the structure of the "Middle Flow". Note that the depth and width here represent the number of the stacked multi-branch blocks and the number of the parallel branches in each block, respectively. A block in Figure 3 means a repeatable structure with a different number of parallel branches, consisting of parallel connections from Xception's original core structure, shown in color in Figure 2. The blocks are then stacked sequentially to different depths. The sub-graph (a) of Figure 3 is the structure of Xception with a block depth of 8 and branch width of 1, while structure 2 in sub-graph (b) has a depth of 4 blocks and width of 2 branches, which is called "XC 4-2", named in the manner of "Xception depth-width". Similarly, sub-graph (c) "XC 2-4" and (d) "XC 1-8" show the structure of 4 branches in 2 block depths and 8 branches in 1 block depth, respectively.

In general, we change the original 8 times repeatable stacked linear structure to different multiples of parallel expansion and depth reduction, with the principle of the total number of core structure constant (=8) to avoid the increase in model parameters. Note that model "XC 1-8" has the same structure as the M_b Xception [37] in which the original 8 times repeated linear structure is changed to 8 branches without linear repeats. The difference is that we are referring to the trade-off between depth and width before and after adopting the attention mechanism. We call the 3 new structures branch expansion networks. A detailed ablation study is conducted on our two datasets and the public dataset, Trashnet [10], to find which of the four structures performs better.

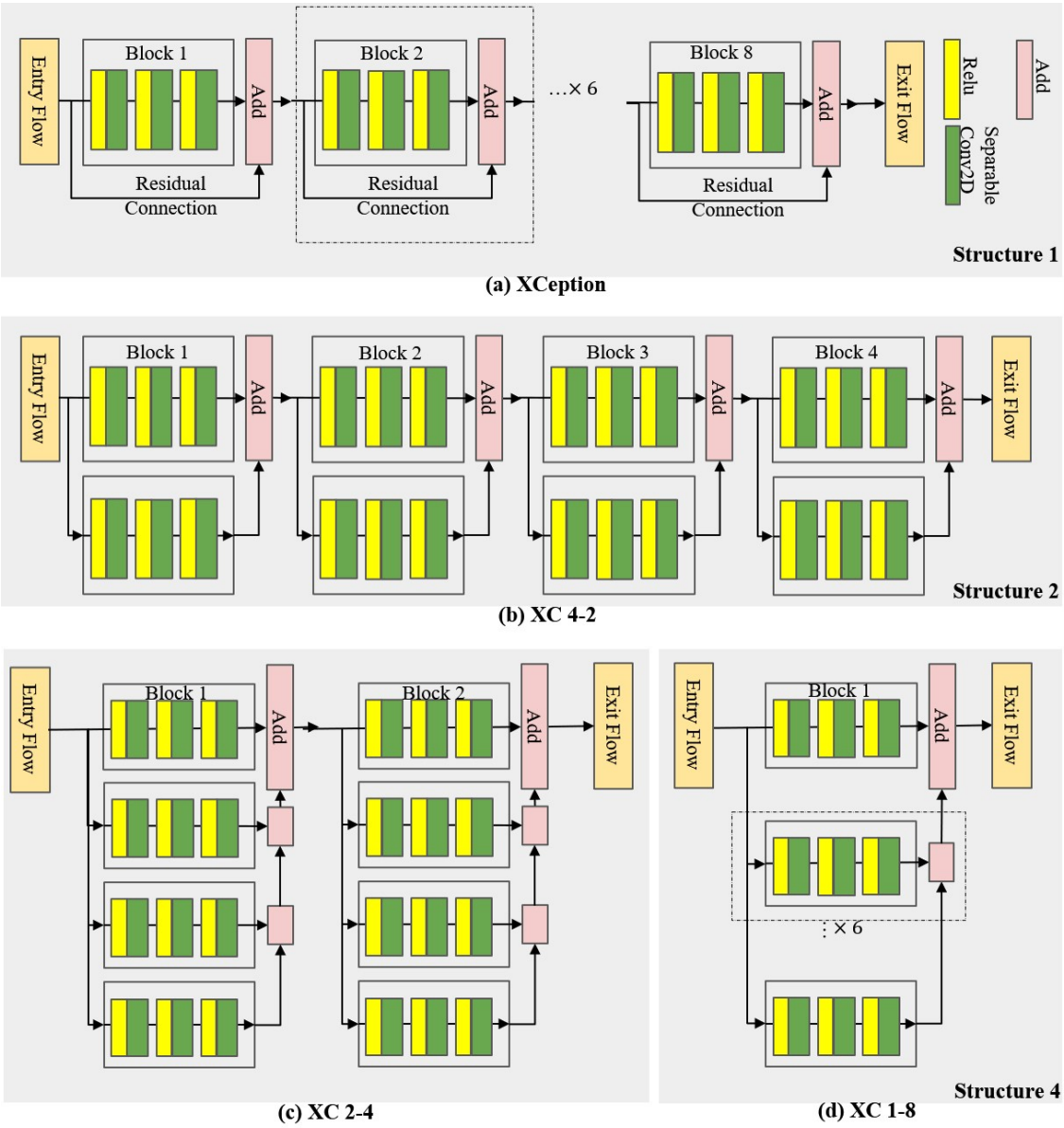


Figure 3. Xception and 3 branch expansion networks. Note that each separable convolution layer is followed by batch normalization (not marked in the figure). (**a**) Xception. (**b**–**d**) Branch extension networks with different depths and widths after core structure recombination, respectively, denoted as "XC Depth-Width". The depth and width here represent the number of the stacked block and the parallel branches in each block, respectively.

3.2.2. ABE Networks and Attention Module

There exist two challenges to our datasets. Since the same object has different states and different objects may be in the same state, such as "held" with hands as the same confusing feature, it can easily lead to confusion between classes. Similarly, the "wearing-state" has the characteristics of a large intra-class gap and a small inter-class gap. Another challenge is that the amount of data is relatively small. We try to find a lightweight method to improve the recognition rate without significantly increasing the number of parameters. Inspired by CBAM [19] and BAM [39], the plug-and-play attention mechanism is a wise choice, which is consistent with our block-stacked-mode network structure.

Figure 4 shows the ABE networks with different structures, which have different depths and widths consisting of different numbers of the multi-branch block with each block followed by an attention module (represent in dotted blue block). We further explain the connection mechanism between the attention module and the multi-branch block in

Figure 5, which is linearly repeatable, according to the depth of the ABE models. To distinguish these attention-based networks, we use a similar naming convention "ABE depth-width" defined in Section 3.2 to name each network as (a) "ABE 8-1", (b) "ABE 4-2", (c) "ABE 2-4" or (d) "ABE 1-8", corresponding to the 4 sub-graphs in Figure 4. In particular, "ABE 8-1" refers to attention-based Xception, which has a single branch with a core structure that repeats 8 times sequentially. THe attention module is placed at the end of each core structure in the "Middle Flow" of Xception.

We introduce our plug-and-play dual-attention module below. Through a large number of comparative experiments, we find that using both channel attention and spatial attention and combining them in the way of parallel residual connection achieves the best results on our datasets. The improved attention module is shown in Figure 6. The parallel connection means that feature F is input into the two attention modules to generate spatial attention and channel attention, respectively, and then the two are concatenated together. Residual connection refers to attaching the input feature to the output feature graph of the attention module to facilitate the learning of the deep network.

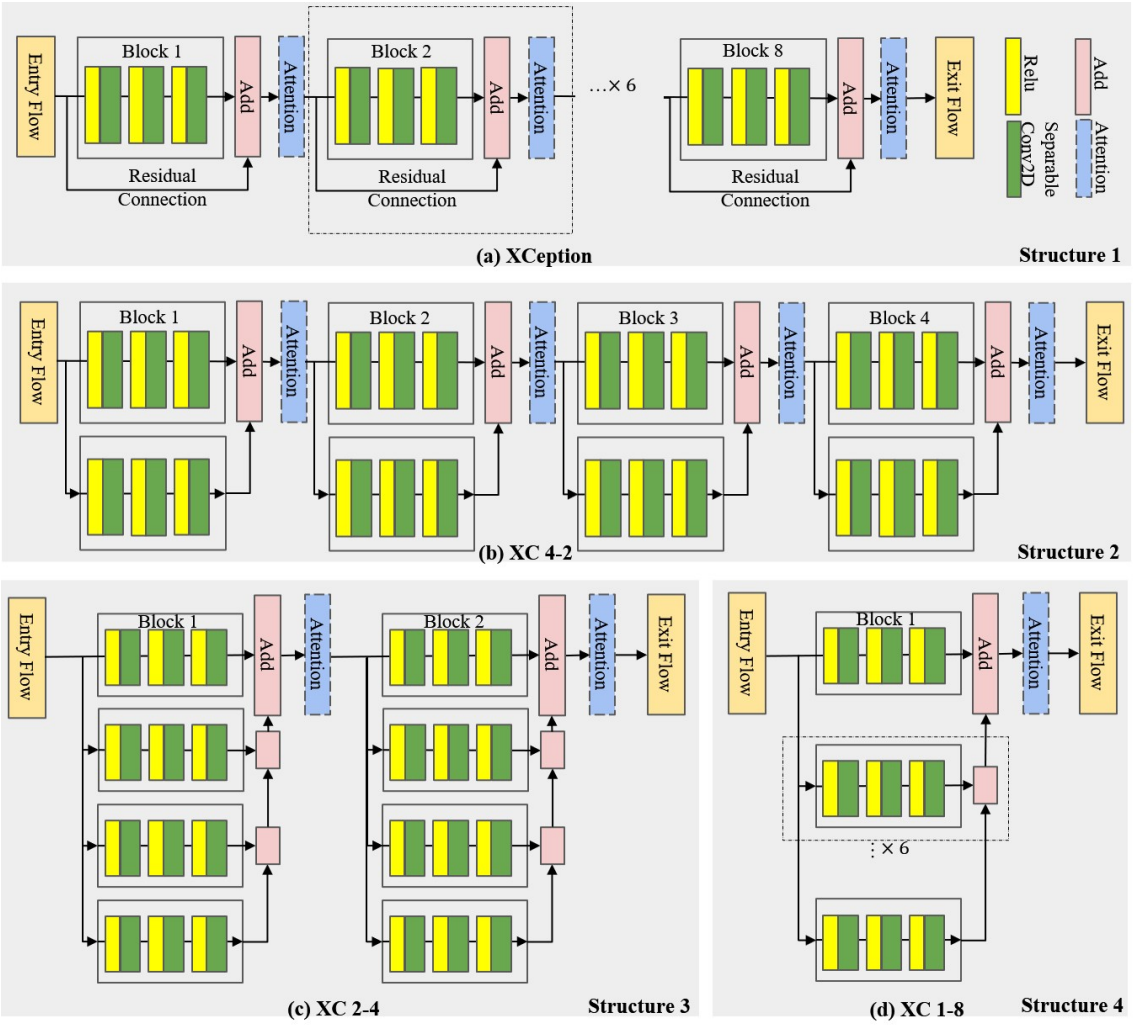


Figure 4. Attention-based branch expansion networks. The attention module is plugged at the end of each multi-branch block. (**a**) Attention-based Xception. (**b**–**d**) "ACE Depth-Width" networks with the same structure of the 3 new "XC" networks.

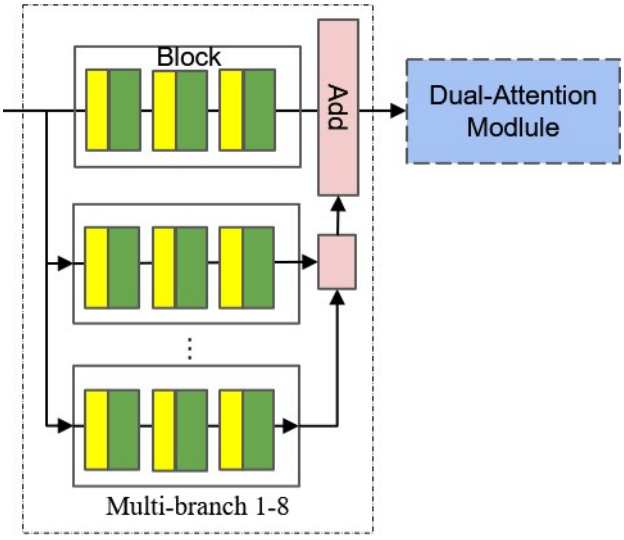


Figure 5. Schematic for the connection between the attention module and a multi-branch block.

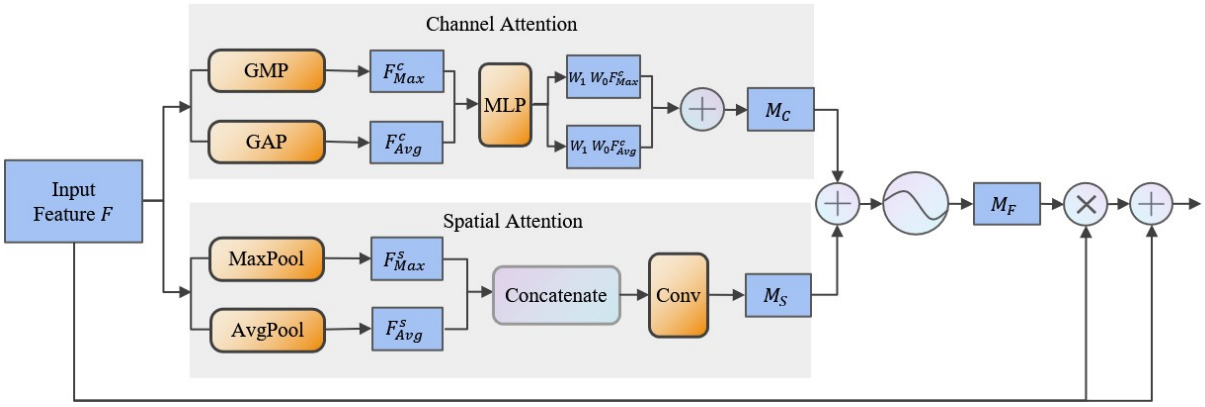


Figure 6. A plug-and-play dual-attention module with parallel residual connection to refine the feature extract by the multi-branch blocks mentioned in Figure 3.

The input feature map $F \in \mathbb{R}^{C\times H\times W}$ from a separable convolution block is fed to the attention module, and a 1D channel attention map $M_c \in \mathbb{R}^{C\times 1\times 1}$ and a 2D spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$ are produced.

We follow the same practice as CBAM to calculate the channel attention map M_c and spatial attention map M_s. The channel attention focuses on "what" is important in the input to obtain the spatial information to obtain the 1D channel descriptor. Hu et al. [41] also proposed to use average pooling to generate channel-wise statistics by aggregating F through spatial dimensions $H \times W$. In addition, Woo et al. [19] believed that the max pool also contains another important clue. So, we adopt the global average pool (GAP) and the global max pool (GMP) simultaneously to shrink the input feature F from $H \times W$ and obtain two channel-wise descriptors F^c_{avg} and F^c_{max}. Then, a multi-layer perception (MLP) capable of learning a nonlinear interaction between channels is used to infer the channel attention map from F^c_{avg} and F^c_{max}. Since the two pools play the same role thus have the same status, the weights of MLP are shared with both of them.

$$\begin{aligned} M_c &= MLP(GAP(F)) + MLP(GMP(F))) \\ &= W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max})) \end{aligned} \tag{1}$$

where GAP and GMP denote the pooling operations, and W_1 and W_0 are learnable parameters of the MLP module that are shared with both F^c_{avg} and F^c_{max}.

The spatial descriptor is also generated from the average pool and max pool operations. Different from the channel-wise statistics, the spatial attention focuses on "where" so that the squeeze input from C to 1 to compute the spatial statistics across the channel dimension is similar to the practice in [42]. Another difference is that the two are concatenated together to generate the spatial descriptor because they both describe features on a plane dimension. A standard convolution layer with the filter size of 7×7 is adopted to encode the spatial descriptor $[F^s_{avg}; F^s_{max}]$ to obtain attention map M_s as follows:

$$\begin{aligned} M_s &= f^{7\times7}([AvgPool(F), MaxPool(F)]) \\ &= f^{7\times7}([F^s_{avg}; F^s_{max}]) \end{aligned} \tag{2}$$

The parallel concatenated dual-attention map M_F and the refined feature F_A is computed as follows:

$$M_F = \sigma(M_c + M_s) \tag{3}$$

where σ is a sigmoid activation function. Equation (3) represents the parallel connection of the two attention maps. Different from the sequential fusion of the two attention maps in CBAM [19], we conduct multiple experiments and prove that the parallel connection shown in Figure 6 is more friendly to our datasets.

$$F_A = F + F \otimes M_F \tag{4}$$

where $\otimes$ denotes element-wise multiplication. Equation (4) gives a residual connection scheme along with the parallel attention following the practice in [38,39] to facilitate the gradient flow and avoid model overfitting.

We apply this plug-and-play dual attention module in the aforementioned 4 structures as illustrated in Figure 4. The attention module is plugged at the end of each multi-branch block, which has the same number of depths of the block. Since the blocks followed by an attention module each has a different arrangement in the 4 structures, the attention module also is used at different locations; therefore, the influence of attention on the model is different. Even if an XC network of a certain depth and width proves to be the best performer, it is not certain whether an ABE network based on that structure is the best performer. So, we still apply the attention module to each of the four structures in the form of convolutional blocks, and conduct detailed experimental comparisons in Section 4.3.

4. Experiments

In this section, we evaluate the performance of the proposed 4 branch expansion structures and 4 ABE networks to evaluate two important issues:

- How do the depth and width of CNNs affect their performance?
- How does the dual-attention module in different numbers and positions affect CNNs of different depth and width?

In order to answer these two questions, we conduct detailed ablation studies in this section by training the 8 networks with 4 structures corresponding to 2 models each ("XC" and "ABE"). Section 4.2 focuses on the first question, while Section 4.3 answers the second question. In addition, in Section 5.1, to illustrate the results of these two issues more intuitively, we visualize the 4 models presented in Figure 3 with different depths and widths as well as the 4 models proposed in Figure 4 with the attention module in different numbers and positions.

4.1. Experiment Settings

We evaluate our proposed models on 3 datasets with a small number of images: one public dataset called Trashnet, and two state datasets that we collected based on our own engineering requirements. As argued in Section 3.1, the challenge with "object-state" and "wearing-state" is the large intra-class distinction and the small inter-class distinction. We try to determine if extending the network's channels could solve these problems and

whether the attention mechanism can refine the key features in order to narrow the intra-class gap and increase the inter-class gap. A data augmentation technique is used in our experiments.

In order to ensure that the experimental results only reflect the performance of the model, the data enhancement settings and the hyperparameters settings used in the experiments are consistent on the same dataset. We use the ImageDataGenerator class from Keras's image preprocessing for image augmentation, and follow the related settings in [37] to facilitate the comparison of performance on the same dataset. We select the appropriate parameters through a large number of experiments and list the hyperparameters settings in Table 3. As for the Trashnet dataset, we divide the dataset to follow the official practice (with 70/13/17 train/Val/test split) and report the accuracy of trash classification on the unseen test sets.

It is worth explaining that the initial learning rate is 0.001 and "decay_rate" represents a multiple of the reduction in the learning rate, calculated by $lr = lr \times decay_rate$, where lr denotes the learning rate. The value of "decay_step" represents the decrease in the learning rate when the model performance has not been improved for many epochs in the past. The minimum learning rate "min_lr" is set as 5×10^{-5}.

Table 3. Hyperparameters settings.

Items	Object-State	Wearing-State	Trashnet
epoch	200	100	500
batch_size	128	128	32
num_classes	14	8	6
learning rate	0.001	0.001	0.001
Optimizer	SGD	SGD	SGD
Momentum	0.9	0.9	0.9
decay_step	10	10	20
decay_rate	0.5	0.5	0.5
min_lr	5×10^{-5}	5×10^{-5}	5×10^{-5}

4.2. Comparison of Branch Expansion Structures

In this experiment, we discuss how the depth and width of the model affect its performance by comparing the accuracy of the four networks proposed in Figure 3 on three different datasets. The results are shown in Table 4 in which we can intuitively obtain the order of the model's performance by longitudinal contrast. For different datasets, we maintain the training parameters of the four models on the same dataset to ensure that the results of different networks only reflect the influence of the structure.

For the "object-state" dataset, the highest accuracy 77.34% comes from the model "XC 1-8",represented in bold, which is 4% higher than Xception and 1.5% higher than the second-best model "XC 2-4". The "wearing-state" dataset gains the highest accuracy of 86% by "XC 1-8", improved from Xception by 3% and 1.6% compared to "XC 2-4". As for the public dataset "Trashnet", the model "XC 1-8" achieves an accuracy of 93.36%, which is improved by 2% compared to Xception. The accuracy of model "XC 2-4" is 1% higher than that of Xception. We can find that the model "XC 1-8" with the depth of 1 block and width of 8 branches performs best on all 3 datasets, followed by the model "XC 2-4", being about 1–1.5% lower. While the model "XC 4-2" and original Xception are about the same performance on the three datasets. The result illustrates that datasets with relatively small amounts of images or obscure characteristics tend to require wider networks rather than deeper ones to learn more detailed channel characteristics.

Table 4. Comparison of the performance (%) between the 4 structures without attention on our proposed two datasets and one common dataset.

Method	Object-State	Wearing-State	Trashnet
Xception	73.44	83.11	91.41
XC 4-2	73.09	83.59	90.62
XC 2-4	75.78	84.38	92.58
XC 1-8	**77.34**	**86.00**	**93.36**

4.3. Comparison of Attention-Based Branch Expansion Structures

In the following experiment, we conduct a longitudinal comparison of the four "ABE Depth-Width" models and a horizontal comparison of the "XC" and "ABE" models in pairs with the same structure. First, the position and quantity of the attention module vary with the structure of the model. Therefore, even if the model "XC 1-8" with a one-block, eight-branch structure performs best, as mentioned above, it is not certain that the corresponding attention model "ABE 1-8" is the best. It is necessary to conduct detailed ablation studies on the attention-based models of the four structures.

Table 5 consists of two parts; the front four rows are the performance of the four models without attention, which are listed here to facilitate horizontal comparisons between the models with the same structure, and the last four rows list the results of the four models after adopting the attention mechanism. It can be found that model "ABE 2-4" performs best on all the three datasets indicated in bold in row 7. We also found that using the attention mechanism does not necessarily improve performance, and may even result in a decline, such as "ABE 8-1" decreased by 3%, compared to the Xception model on the "wearing-state" dataset. Coincidentally, the "ABE 1-8" model, after adding the attention module to "XC 1-8", which performs best among the four attention-free models, decreases on all three datasets and performs worst among the four attention-added models. For structure 2, in contrast to "ABE 4-2" and "XC 4-2", the performance is slightly improved after using the attention module. Therefore, for the deepest original Xception and the widest network "XC 1-8", there is no need to use the attention mechanism. For "XC 2-4", with 2 stacked blocks, each consisting of 4 parallel branches, the attention module after each block does refine features extracted by the block with an improvement of about 3–4%. The best attention-added model "ABE 2-4" is about 2% higher than the best attention-free model "XC 1-8" on the three datasets. This conclusion is further illustrated in the visualization results in the next sub-section.

Table 5. Longitudinal comparison of the 4 "ABE Depth-Width" models and horizontal comparison of models in pairs with the same structure on the 3 datasets.

Method	Object-State	Wearing-State	Trashnet
Xception	73.44	83.11	91.41
XC 4-2	73.09	83.59	90.62
XC 2-4	75.78	84.38	92.58
XC 1-8	**77.34**	**86.00**	**93.36**
ABE 8-1	73.24	80.03	91.34
ABE 4-2	76.56	83.43	91.88
ABE 2-4	**79.69**	**88.28**	**94.53**
ABE 1-8	71.09	82.79	91.80

The above conclusions explain that the performance of the linear stacked block models with different widths and depths is not consistent before and after adding block attention modules. The challenge datasets with relatively few images and few key features call for widening the network, instead of deepening it. While using the attention mechanism to

further refine features, we need to balance the depth and width of the network to achieve the perfect balance with the attention module. We verify the above conclusions through visualization results in the next part and exhibit the parameters and computation of the eight networks on the Trashnet dataset to illustrate the additional overhead of parameters and computation with the addition of attention modules. It matters which model we ultimately choose as the best practice balancing the performance and overhead.

5. Discussion

5.1. Visualization and Overhead

In addition to the above quantitative comparison between longitudinal and lateral, we also compared the performance of the model intuitively through visualization results. Figure 7 shows three groups of Grad-CAM visualization results for the eight networks. Grad-CAM [22] represents gradient-weighted class activation mapping that uses the gradient information of the last convolution layer flowing into the CNN to allocate important values for each neuron. This is a technique to visualize what is going on inside CNN by highlighting the learned important regions in a heatmap.

We selected nine representative images from test sets of the three datasets to show the visualization results. For the first four "XC depth-width" models without attention, it can be seen that the key regions extracted by "XC 1-8" are more accurate and more complete, which is consistent with the conclusions of the previous Section 4.2. For example, the "XC 1-8" is most accurate in determining the position of "newspapers", "metal cans" and "glass bottles". We observe the following four models with the attention module plugged in and find that the model "ABE 2-4" has the best characteristic expression. It can be seen that "ABE 2-4" has the best attention ability for the "Lw_s" and "Sd_l" classes in columns 1 and 2, even better than "XC 1-8". The results are consistent with those in Section 4.3. It is worth mentioning that the advantage of "ABE 2-4" lies in the relatively accurate position information, such as "Lw_s" (column 1) and "paper" (column 7), which can see the boundary close to the original image. The qualitative visualization results are consistent with the quantitative results in Tables 4 and 5.

In addition, we also discuss the influence of different model structures and the addition of attention on the number of model parameters and computation speed. We list the accuracy, the number of parameters, and single batch training time of the eight networks on the Trashnet dataset in Table 6. The first four models have the same number of parameters and training time since we just reorganize the eight linearly stacked core structures of Xception instead of deepening the stacking further or adding additional branches so that the number of layers and the number of channels in each layer of "XC 4-2", "XC 2-4" and "XC 1-8" is exactly equal to Xception. As for the ABE models, it can be seen from Figure 3 that the number of attention modules follows the depth of the model, which is a stacked-block structure, resulting in different parameters and training times for the four ABE models.

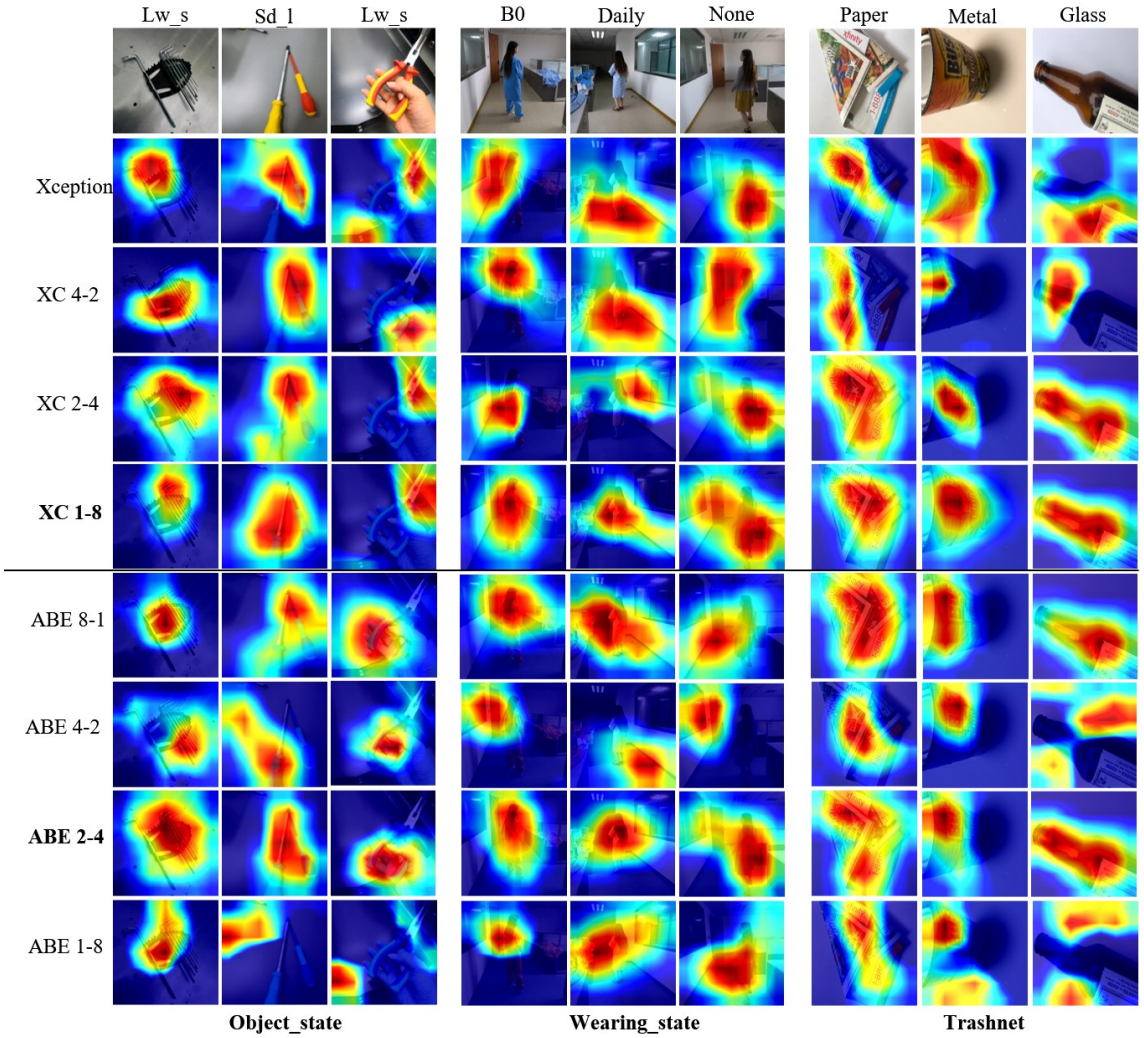


Figure 7. The Grad-CAM visualization results of 8 models. We compare Xception with 7 other models to see how different widths and depths affected the model's performance, and how the attention module fits the widths and depths of the backbone.

Table 6. Comparison of accuracy, number of parameters and single batch training time of the 8 networks on Trashnet dataset.

Method	Accuracy (%)	Parameters (M)	S-B Train Time (ms)
Xception	91.41	83.8	147
XC 4-2	90.62	83.8	147
XC 2-4	92.58	83.8	147
XC 1-8	93.36	83.8	147
ABE 8-1	92.34	90.5	173
ABE 4-2	91.88	87.2	163
ABE 2-4	94.53	85.5	160
ABE 1-8	91.80	84.7	155

Our attention module is a lightweight plug-and-play module that makes the exceeding amount of parameters and training time negligible in the premise of improving model performance. In general, the best model "ABE 2-4" with 2 M (2%) more parameters than Xception, and takes only 13 ms more training time for a single batch (32 images). We compare the performance of our "ABE 2-4" model with the SOTA technologies in Section 5.2 on the common dataset, Trashnet.

5.2. Comparison with State-of-the-Art Results

We verify the superiority of the "ABE 2-4" model on the public dataset Trashnet by comparing with the following relatively new technologies of recent years, including traditional machine learning methods and deep learning methods. We visually show the level of accuracy in the form of a statistical histogram graph in Figure 8. Our proposed model, ABE 2-4, is shown in the first line, while other deep learning methods are shown in orange, and the traditional machine learning methods are shown in blue.

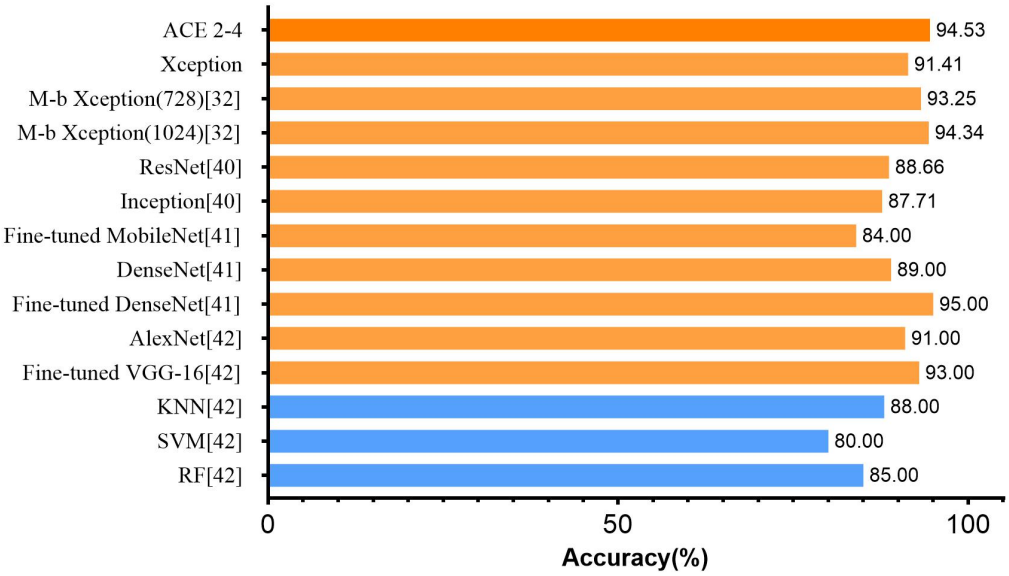


Figure 8. The statistical histogram graph of the comparison between the proposed "ABE 2-4" model and relatively new technologies in recent years.

ResNet [23] gains an accuracy of 88.66%, and Inception [23] gains 87.71% accuracy, while Xception achieves 91.41% accuracy in our experiment. In 2020, The M_b Xception [37] modified from the Xception improved by 3% in which the number of convolution channels increased from 728 to 1024; their 728-channel model achieved 93.25%. The DenseNet [21] achieves 89% and the fine-tuned one achieves the highest accuracy of 95% among deep learning methods by fine-tuning from the pre-trained weights on the ImageNet data. The above two are similar to our results. Bernardo et al. [43] compared the VGG-16 with traditional methods and reached an accuracy of around 93% on fine-tuned VGG-16. Among the traditional machine learning technologies, the KNN performs best with an accuracy of 88%. It can be seen that the precision of fine-tuning a model can be greatly improved when the premise is still big data.

We show the confusion matrix of the original Xception and the final "ABE 2-4" model in Figure 9. The vertical axis represents the true label, the horizontal axis represents the predicted label, and the diagonal element is the perception of pictures predicted to the correct class. The confusion matrix of "ABE 2-4" in Figure 9 shows consistent improvements in classification performance on each class; in particular, "Glass", "Metal", and "Trash", which are difficult to predict, demonstrate the stability of the model on a small sample dataset.

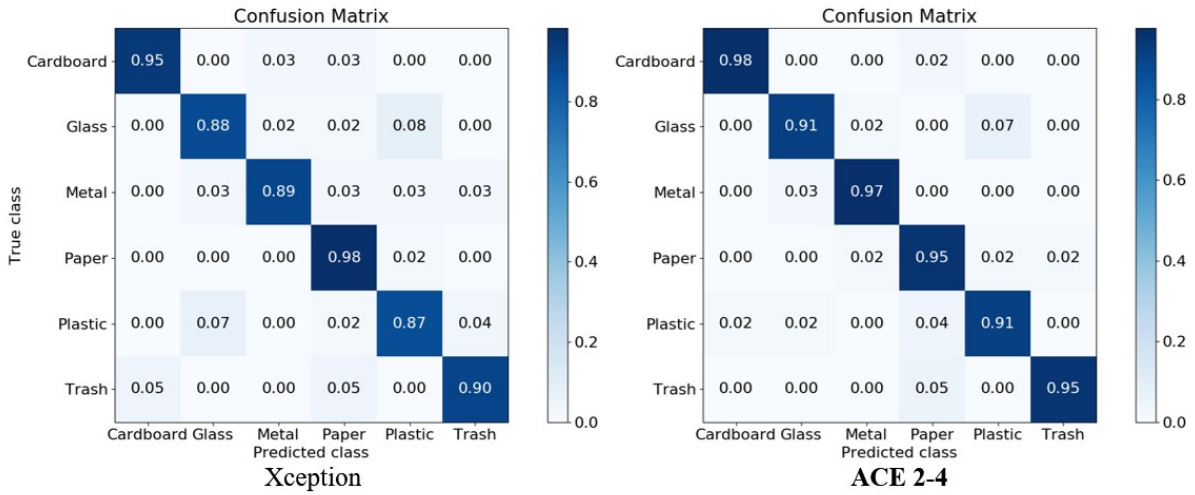


Figure 9. Confusion matrix of the original Xception and the final "ABE 2-4" with 2 linear blocks followed by a dual-attention module each, with 4 parallel branches in each block.

6. Conclusions and Future Work

In this work, we collect two new datasets from our digital assembly test center (DTAC) for the monitoring of laboratory technical states. These two datasets have a small amount of data; the change of illumination, background, type, etc., result in a large intra-class difference and smaller inter-class difference. We propose a generic, attention-based branch expansion network (ABE) by reorganizing and modifying the core structure of Xception and embedding the dual-attention module with a parallel residual connection. The ABE networks consist of linearly stacked blocks with parallel branches, each of which is followed by a plug-and-play attention module with the negligible overhead of parameter and computation. We conduct detailed ablation studies and carry out full horizontal contrast and vertical contrast. The "ABE 2-4" model with two stacked blocks, each with four branches, performs best among the four attention-added structures with different depths and widths. We reach an accuracy of 94.53%, which is comparable to the SOTA technologies on Trashnet.

Our method has a wide application prospect. In addition to the classification of technical states in the laboratory, it can also be applied to dress pattern classification, trash classification, medical diagnosis, etc. However, there are still some limitations. First of all, the category and quantity of datasets are small. In addition, this is a single-image classification model, which is not enough to be applied in real multi-target scenes. To further improve the model performance, a trained YOLO model can be used as a front end in practical application, and the generated bounding boxes can serve as the input of our model. Our dataset acquisition method, Fly_arounds video, fits the end-to-end requirement.

Author Contributions: Conceptualization, S.S.; methodology, Y.Z. and S.S.; software, Y.Z., H.L. and L.L.; validation, Y.Z. and G.L.; formal analysis, Y.Z. and S.S.; investigation, Y.Z. and G.L.; resources, S.S.; data curation, Y.Z. and D.L.; writing original draft preparation, Y.Z.; writing review and editing, Y.Z. and S.S.; visualization, Y.Z.; supervision, S.S.; project administration, S.S. and Y.Z.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study, because this paper does not involve human or animal research. It studies scene understanding and image classification in the computer vision field; the wearable state classification is one of the image classification tasks, not medical pathology research.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: [https://github.com/garythung/trashnet (accessed on 6 June 2021)].

Acknowledgments: The authors would like to acknowledge Gary Thung and Mindy Yang for making their datasets available.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABE	Attention-Based Branch Expansion Network
SVM	Support Vector Machine
RF	Random Forest
KNN	K-Nearest Neighbor
BAM	Bottleneck Attention Module
CBAM	Convolutional Block Attention Module
CNN	Convolutional Neural Network
GAP	Global Average Pooling
GMP	Global MAX Pooling
Grad-CAM	Gradient-weighted Class Activation Mapping

References

1. Chen, J.B.; Zhai, G.F.; Wang, S.J.; Liu, Y.; Wang, H.Y. Factors affecting characteristics of acoustic signals in particle impact noise detection for aerospace devices. *Syst. Eng. Electron.* **2013**, *35*, 889–894.
2. Guofu, Z.; Jinbao, C.; Qiuyang, L. Detecting loose particle signals in multichannel recordings with transductive confidence predictor. *Trans. Inst. Meas. Control* **2015**, *37*, 265–272. [CrossRef]
3. Alayrac, J.B.; Laptev, I.; Sivic, J.; Lacoste-Julien, S. Joint discovery of object states and manipulation actions. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2127–2136.
4. Aboubakr, N.; Crowley, J.L.; Ronfard, R. Recognizing manipulation actions from state-transformations. *arXiv* **2019**, arXiv:1906.05147.
5. Farhadi, A.; Endres, I.; Hoiem, D.; Forsyth, D. Describing objects by their attributes. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1778–1785.
6. Duan, K.; Parikh, D.; Crandall, D.; Grauman, K. Discovering localized attributes for fine-grained recognition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3474–3481.
7. Li, Y.L.; Xu, Y.; Mao, X.; Lu, C. Symmetry and group in attribute-object compositions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11316–11325.
8. Wang, H.; Pirk, S.; Yumer, E.; Kim, V.G.; Sener, O.; Sridhar, S.; Guibas, L.J. Learning a Generative Model for Multi-Step Human-Object Interactions from Videos. *Comput. Graph. Forum* **2019**, *38*, 367–378. [CrossRef]
9. Isola, P.; Lim, J.J.; Adelson, E.H. Discovering states and transformations in image collections. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1383–1391.
10. Yang, M.; Thung, G. Classification of trash for recyclability status. *Cs229 Proj. Rep.* **2016**. Available online: https://pdfs.semanticscholar.org/c908/11082924011c73fea6252f42b01af9076f28.pdf (accessed on 5 August 2021).
11. Pang, G.; Shen, C.; Cao, L.; Hengel, A.V.D. Deep learning for anomaly detection: A review. *ACM Comput. Surv.* **2021**, *54*, 1–38. [CrossRef]
12. Montúfar, G.; Pascanu, R.; Cho, K.; Bengio, Y. On the number of linear regions of deep neural networks. *arXiv* **2014**, arXiv:1402.1869.
13. Chen, L.; Wang, H.; Zhao, J.; Papailiopoulos, D.; Koutris, P. The effect of network width on the performance of large-batch training. *arXiv* **2018**, arXiv:1806.03791.
14. Shang, W.; Sohn, K.; Almeida, D.; Lee, H. Understanding and improving convolutional neural networks via concatenated rectified linear units. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2217–2225.
15. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
17. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
18. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
19. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.

20. Fathi, A.; Rehg, J.M. Modeling actions through state changes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2579–2586.
21. Aral, R.A.; Keskin, Ş.R.; Kaya, M.; Hacıömeroğlu, M. Classification of trashnet dataset based on deep learning models. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 2058–2062.
22. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
23. Ruiz, V.; Sánchez, Á.; Vélez, J.F.; Raducanu, B. Automatic image-based waste classification. In Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation, Almería, Spain, 3–7 June 2019; pp. 422–431.
24. Lin, Y.; Lv, F.; Zhu, S.; Yang, M.; Cour, T.; Yu, K.; Cao, L.; Huang, T. Large-scale image classification: Fast feature extraction and svm training. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1689–1696.
25. Chaganti, S.Y.; Nanda, I.; Pandi, K.R.; Prudhvith, T.G.; Kumar, N. Image Classification using SVM and CNN. In Proceedings of the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 13–14 March 2020; pp. 1–5.
26. Zhao, C.; Zhao, H.; Wang, G.; Chen, H. Improvement SVM classification performance of hyperspectral image using chaotic sequences in artificial bee colony. *IEEE Access* **2020**, *8*, 73947–73956. [CrossRef]
27. Guo, B.; Gunn, S.R.; Damper, R.I.; Nelson, J.D. Customizing kernel functions for SVM-based hyperspectral image classification. *IEEE Trans. Image Process.* **2008**, *17*, 622–629. [CrossRef]
28. Sheykhmousa, M.; Mahdianpari, M.; Ghanbari, H.; Mohammadimanesh, F.; Ghamisi, P.; Homayouni, S. Support vector machine vs. random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 6308–6325. [CrossRef]
29. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Wang, R. Efficient kNN classification with different numbers of nearest neighbors. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 1774–1785. [CrossRef]
30. Wang, P.; Fan, E.; Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognit. Lett.* **2021**, *141*, 61–67. [CrossRef]
31. KOUSTUBH. ResNet, AlexNet, VGGNet, Inception: Understanding Various Architectures of Convolutional Networks. 2018. Available online: https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception (accessed on 5 August 2021).
32. Yilmazer, R.; Birant, D. Shelf Auditing Based on Image Classification Using Semi-Supervised Deep Learning to Increase On-Shelf Availability in Grocery Stores. *Sensors* **2021**, *21*, 327. [CrossRef]
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
36. Nicholus, M.; Claudio, P.; John, B.; Alfred, S. Detection of Informal Settlements from VHR Images Using Convolutional Neural Networks. *Remote Sens.* **2017**, *9*, 1106.
37. Shi, C.; Xia, R.; Wang, L. A Novel Multi-Branch Channel Expansion Network for Garbage Image Classification. *IEEE Access* **2020**, *8*, 154436–154452. [CrossRef]
38. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
39. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
40. Li, H.; Qiu, K.; Chen, L.; Mei, X.; Hong, L.; Tao, C. SCAttNet: Semantic segmentation network with spatial and channel attention mechanism for high-resolution remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 905–909. [CrossRef]
41. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
42. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv* **2016**, arXiv:1612.03928.
43. Costa, B.S.; Bernardes, A.C.; Pereira, J.V.; Zampa, V.H.; Pereira, V.A.; Matos, G.F.; Soares, E.A.; Soares, C.L.; Silva, A.F. Artificial intelligence in automated sorting in trash recycling. In Proceedings of the Anais do XV Encontro Nacional de Inteligência Artificial e Computacional, Sao Paulo, Brazil, 22–25 December 2018; pp. 198–205.