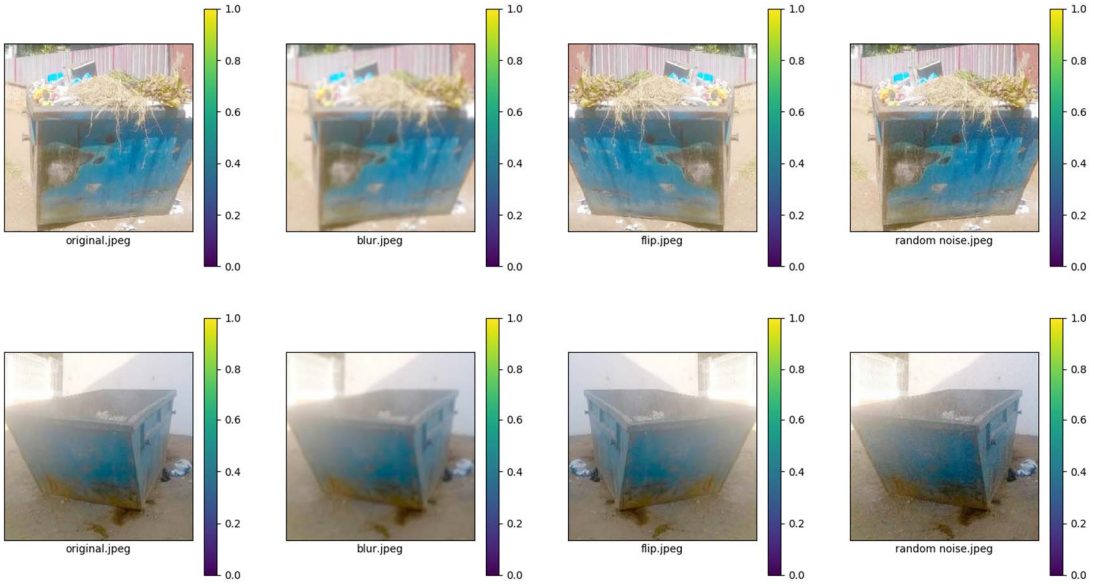


Fig. 2**.** Some images of the built dataset with augmented data. The first row is composed by container images labeled as full and the second row is composed by container images labeled as not full.

### *B. Building the models*

Tensorflow with keras API was used to build models by taking into account the following DCNNs architecture: ResNet34, ResNet50, Inception_v4 and DarkNet53. The RetinaNet detention model uses a Feature Pyramid Network (FPN) backbone on top of a feed-forward ResNet architecture [10]. The Yolo_v3 detention model uses DarkNet53 for performing feature extraction [11].

### *C. Training and Testing*

One of the problem with evaluating a model is that it may demonstrate adequate prediction capability on the training data, but might fail to predict future unseen data and cross-validation is a procedure for estimating the generalization performance in this context [12].

The procedure chosen to train and test the models was Repeated K-Fold Cross-Validation. "In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning" [12]. As in k-fold cross-validation, only k estimates are obtained, a commonly used method to increase the number of estimates and to obtain less biased performance estimation is to run k-fold cross-validation multiple times. The data is reshuffled and re-stratified before each round [12]. In this study the 5-fold cross-validation was run 6 times, which gave 30 points of estimation for each metric, namely: PR-AUC, F1, PRECISION, RECALL and ACCURACY [11].

The mean and sample standard deviation for each metric were calculated, as the points of estimation is equal to 30 we built a confidence interval (CI) with a level of confidence $\boldsymbol{c}$ = **0.95.** We used *ADAM* [13] optimizer with learning rate of 1e-4 during all training process and sparse_categorical _crossentropy as loss function.

### *D. Finding the correlation between depth and ACC*

"The most straightforward way of improving the performance of deep neural networks is by increasing their size. This includes both increasing the depth – the number of network levels – as well as its width: the number of units at each level" [14]. In order to find out if this happen on the chosen models, the correlation between the model depth and ACC was calculated.

## IV. RESULTS AND DISCUSSION

All the experiments of this study were performed using TensorFlow version 1.13.1 with Keras API on a hardware with 8 GB of RAM and 7$^{th}$ generation i7 dual core processor with 2.90 GHZ. The sklearn library was used to compute the performance of the models. The performances of the models are shown in Table I.