Classifying Scrap Metal on the Edge

Chitra S. Agastya
chitra.agastya@berkeley.edu

Bradley Nott
bnott@berkeley.edu

Marcus A. Streips
mastreips@berkeley.edu

Stanley Ye
stanley.ye@berkeley.edu

Abstract

A method of classifying scrap metal using a deep learning neural network running on an edge computing device is presented. Multiple convolutional neural networks using both transfer learning and training from scratch were assessed using raw photo data provided by a scrap metal processing plant. This proof-of-concept demonstrates that reducing the turnaround time for validating scrap metal shipments through computer vision IoT technology is possible.

1 Introduction

Recycling is a high-cost, low-margin business. Machines must sort both quickly and accurately, conflicting objects that must be managed carefully. Current machine sorting accuracy is about 90%. The remaining 10% is recovered by manual hand sorting on-site at U.S. plant and/or in Asia sorting facility (Figure 1).

Figure 1: Hand-Sorting of Scrap Metals

The turnaround time for validating a lot of scrap metal is 5 months. This is due to the standard business requirement of sending a 50K lbs test lot to metal smelting operations in India and China via sea freight. Once the test lot arrives at its destination, it is validated via hand-sorting before the authorization of the remaining shipment is made. Therefore, there is a need to be able to validate shipments at various collection centers and ports of entry via automated means to reduce the cycle time of a typical transaction.

Through collaboration with Alumisource, a scrap metal processing business based near Pittsburgh, PA, we received and took photographs of samples from five different types of metal: aluminum, brass, copper, stainless steel, and zinc. Utilizing off-the-shelf, deep learning techniques, we built an image classifier that exceeded 90% accuracy in identifying metal type. These promising results demonstrate the potential of using image classification to replace manual sorting, reducing the overall cycle time of the typical scrap metal transaction.

2 Background

The worldwide scrap metal recycling market in 2019 was US$ 280 billion (EUR 250 billion) and is expected to grow 3% by 2024 to US $ 340 billion in revenue. The US scrap metal recycling industry alone is worth almost US$ 27 billion, processing 110 million tons of recyclables annually[1].

The global metal recycling market is segmented on the basis of metal type, end-user industry, and geography. Based on metal type, the market is segmented into ferrous metals and non-ferrous metals. By end-user industry, the market is classified into automotive, packaging, shipbuilding, industrial machinery, electronics and electrical equipment. Building and construction have generated the highest revenue owing to increased disposal of building waste into the landfills. Geographically, the market is segmented into North America, Europe, Asia-Pacific, Latin America, Middle East and Africa. Asia-Pacific has contributed the highest revenue in 2014 and is expected to dominate

the market for the near foreseeable future[2].

The metal recycling market includes various stakeholders such as scrap traders, recycling companies, end-users and consumers. The scrap traders and dealers are suppliers whereas the end-user industries are the buyers of the market, including mills and smelting plants. Typical end-user industries are automotive, packaging, shipbuilding and building construction. Recycling companies, such as Alumisource, receive the scrap metal from scrap traders where it is sorted, shredded and packaged for shipment, primarily to markets in China and India.

Currently, recycling companies receive metals in mixed lots where it is stored for processing in large, 300K sq ft warehouse facilities. The scrap metal is then put on a conveyer belt where it is separated using a number of different technologies such as magnets (for ferrous metals), x-rays and sink-float processes (for non-ferrous alloys) and air guns (trash) as well as other, sensor-based sorting systems that use camera technology[3]. The practice of manual sorting is still considerably widespread as it results in high purity levels. However, manual sorting is cost intensive in developed countries and should therefore only serves as a final check for quality control purposes.

Not all metals can be sorted manually, as their difference is not discernible visibly. Gray metals can only be sorted to a certain extent, and it can be difficult for stainless steel to be separated visually at all. To tackle this problem, recycling plants use the sink-float process to separate materials of different densities. However, this process can requires large quantities of water and expensive additives, such as ferro-silicon and requires expensive, specialized machinery with limited separation efficiency.

The use of sensor-based sorting systems represents an economic and alternative to manual or density sorting, but has its own limitations. Sorting by color using a color-detecting camera is one option; however, only scrap metals that are clearly distinguishable by color, such as copper and brass, can be separated. All gray heavy metals, such as zinc, lead, nickel and stainless steels, may not be sortable based on color alone. Sorting using a color camera also can be affected negatively by surface contamination and color impurities. Another sensor-based technologies are X-ray transmission (XRT) and X-ray fluorescence (XRF). The transmission technology helps detect density differences (similar to how an X-ray detects bones). In an XRF sorting unit, an X-ray tube emits X-rays, thereby "exciting" the metal piece. Depending on the composition of the metal piece, the radiation emitted from the "excitation" represents the composition of the metal. The most important benefit of XRF technology is the metal sorting regardless of color and surface contamination. Compared with camera technology, dirty and uncharacteristic color pieces of copper and brass, for example, can be accurately detected and sorted. Likewise, it is possible to detect metals of the same or similar color separately and to sort them. The biggest drawback of this technology is its cost and maintenance.

Once the metal is separated, it is shredded into relatively homogeneous pieces using a $5/8$ inch screen, resulting in pieces that are $5/8$ in. thick an between 2 in. and 5 in. in length. It is this shredded metal that was randomly sampled and provided to our team for analysis. The shredded metal is then shipped in 50K lbs. containers for export to international buyers.

3 Methods

3.1 Data

In this project, we decided to build an original data set from scratch for several reasons. We found that publicly available images of scrap metals are not available in the quantity, quality, and variety needed to build a data set that could support our research objectives. We also did not find any complete public data sets containing an appropriate collection of such images. The closest research project that we found with a similar objective to our own is known as TrashNet[4]. That project was an effort by a pair of Stanford University students to train a classifier to recognize items that are often manually sorted at a general purpose recycling facility (e.g., glass, paper, cardboard, etc.). However, while it addressed a similar business problem, the images associated with that research were not suitable for our purposes.

We concluded that the intended use case for our classification model is fairly unique: we required close-up images of individual metal samples from a particular phase of the scrap metal sorting sequence. Specifically, we needed samples that had already been processed by an industrial scrap metal shredding machine. Scrap metal sam-

ples in this specific physical condition have important features (e.g., bends, folds, twists, cracks, etc.) that are essential for training the classification model we had in mind, and these characteristics are not always present in publicly available images of scrap metals. Therefore, in order to train a model that could perform well in the type of environment where it would be deployed, we obtained a representative assortment of scrap metals sampled directly from the Alumisource sorting facility.

The raw sample set contained 500 total pieces, and consisted of 100 samples from each of five classes: aluminum, stainless steel, zinc, copper, and brass. Samples varied in size (approx. 1-8 square inches) with the aluminum and stainless steel classes containing the largest samples, copper and brass containing medium-sized samples, and zinc with the smallest. Images of individual samples were captured with the aid of a photo light box in order to maintain consistent neutral backgrounds and lighting conditions. Individual samples were photographed so that our classification results would help explain whether the variance that exists across the five scrap metal types is sufficient to make meaningful classification decisions. This frames the classification problem in terms of image classification, rather than object detection. While it is certainly possible to build a data set suitable for object detection, the color similarity of the five scrap metal classes would make it challenging to maintain accuracy during the object labeling task. However, if a trained model can perform well on images of individual samples, the results could warrant construction of a more complicated object recognition data set.

It is important to explain that the types of metals involved in this study have some inherent natural color differences, but many of our samples exhibited less of their natural color distinctness because they were dirty (i.e., low purity). While we had no expectation that raw scrap metal samples should be similar in appearance to finished products made from the same material (e.g., finished brass is shiny and yellow), we acknowledge that this factor could have an undesirable influence on the performance of a classification model.

In contrast, the shapes of the samples proved to be a much better source of variability across classes. The shapes themselves are features, which are in part a function of how a particular metal commonly responds to the shredding process. If a sample had a unique three-dimensional shape, multiple images of that sample were included in the data set as long as these additional images were indicative of unique features. If a sample only had two unique aspects, such as the top and bottom of a flat piece of metal, only two images of the sample were included. In this way, care was taken to ensure the data set was not intentionally biased by including seemingly identical images of the same object. We feel these design decisions are appropriate because it better represents the variety of object orientations that would be encountered on a moving conveyor belt. In a sorting facility a camera will only get a brief look at a passing object, and a robust classification model should have the flexibility to recognize types of metal regardless of how their random aspect makes them appear on camera.

With this strategy we assembled a collection of images to support training our models. Samples from each class are presented in Figures 2.

Ultimately, we have 1383 raw images of all metals combined. Of these 27% are aluminum, 13% copper, 21% stainless steel, 17% zinc and 22% brass. To ensure that we have equal distribution of data for each metal category, we randomly split each metal category with a 60/20/20 stratified split for training, validation and test data. Table 1 gives the number of images of each metal overall and the splits used for training and validation. Because we have a small dataset, we enable inline image augmentation to increase the number of data images and add generalization to our training. Although, the classes were slightly imbalanced for zinc and copper, our models did not seem to have trouble classifying these categories, so we opted not to resample the data.

Metal	**# of Images**	**Train**	**Val**	**Test**
Aluminum	369	221	74	74
Copper	178	106	36	36
Stain Steel	299	178	60	61
Zinc	234	140	47	47
Brass	303	181	61	61
Total	1383	826	278	279

Table 1: Data Profile

Figure 2: Samples of Metals

3.2 Data Augmentation

In order to generalize the model, we leveraged the PyTorch inline data augmentation tool, ImageDataGenerator. The tool allows us to simulate the variability we would expect in lighting and perspective on the factory conveyor belt. For example, from an original image (Figure 3) we can use image rotation, scaling and shift to simulate variability of perspective (Figures 4, 6 and 7 respectively). We can use image brightness to simulate different lighting conditions (Figure 5).

Figure 3: Original Raw Image

3.3 Modeling

In this report, we evaluate different deep learning techniques and models to classify scrap metal into one of five categories: aluminum, stainless steel, zinc, copper and brass. Our goal is to see which technique and model performs the best based on the following criteria (1) size of the model (2) performance on test data using metrics such as loss, validation accuracy, f1-score etc. Unlike a well-defined set of objects like animals or people, our data comprises images of white and red scrap metals that have already been through a shredder. These objects are very small and don't have a characteristic shape that could help with identifying the metal. So attributes like color, cracks and warps on a metal surface etc. become potentially important in helping with classification. We look at two main approaches: transfer learning and training from scratch. Our generic strategies for training are outlined in Figure 8[5]. To have a common baseline for evaluation, we use Keras framework to build all of our models.

3.3.1 Transfer Learning

Transfer learning is a popular technique in deep learning where a pre-trained model developed in one setting is re-purposed in a different setting. Even though our metal scraps don't have a characteristic shape, we see that transfer learning is quite effective in classifying the metal scraps. In our initial study we use pre-trained models from VGG-16 as well as ResNet-50. We freeze the convolutional base and replace the original classifier with a fully connected classifier that is designed to pick from one of the five metals (Figure 9). We use the Adam optimizer as it gives a more favorable performance over the RMSProp optimizer. The ResNet-50 model gives us better performance than VGG16 and has a much smaller memory footprint (Table 2). So we proceed with ResNet-50 for further training with fine-tuning and inline image

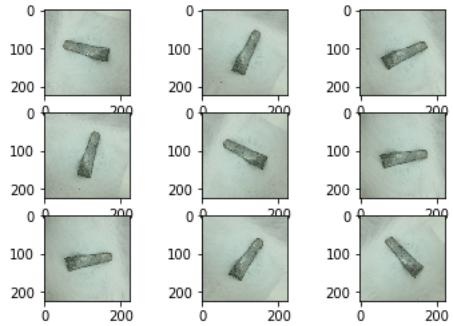

Figure 4: Augmented Data (Rotation)

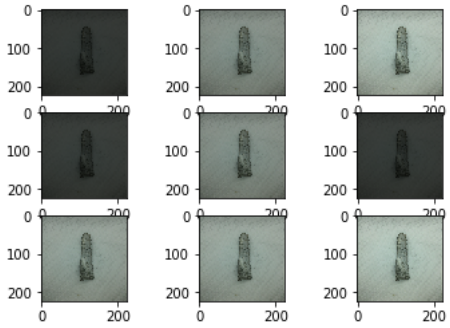

Figure 5: Augmented Date (Brightness)

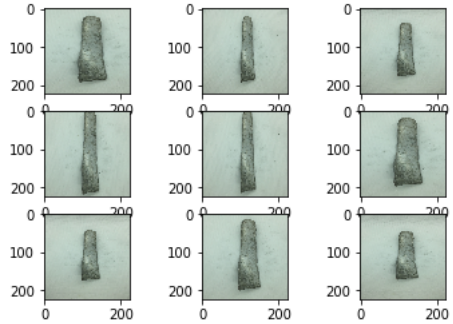

Figure 6: Augmented Data (Scaling)

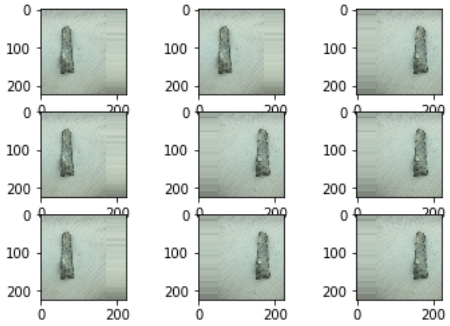

Figure 7: Augmented Date (Shift)

Metrics	VGG-16	ResNet-50
val accuracy	0.68	0.89
loss	1.1	0.39
model size (MB)	300	90

Table 2: Results: Vgg-16 vs ResNet-50

augmentation.

Deep learning models are said to learn hierarchical feature representations where feature computations move from generic to more specific as we move from initial layers to final layers. The features computed in the convolutional base, which is closer to the input, are considered more generic while the features computed in the layers closer to the classifier are supposed to be more specific to the dataset at hand. For fine tuning the model, we take the features extracted from the convolutional base and then feed it into a global average pooling layer. The output of this is again fed into a couple of fully connected and dropout layers before feeding it into a fully connected classifier (Figure 9). We train this fine-tuned model using an Adam optimizer on our raw images with no image augmentation. This boosts our validation accuracy from 0.89 to 0.92. However, our model size goes up to 99MB because of a deeper network.

3.3.2 Training from Scratch

Our data is very different from the original ImageNet data that the pre-trained ResNet-50 model was trained on. Because of this reason, we also consider training the entire model from scratch and use CIFAR 10 and ResNet models for our experiments.

CIFAR 10

We started our from scratch modeling using a simple CNN based on the off-the-shelf CIFAR10 model in Keras[6]. The CIFAR10 model architecture includes 4 convolutional layers, with interspersed max-pooling and drop-out layers (Figure 10). The model takes a 244 x 244 x 32 input matrix (corresponding to the 244 x 244 pixel image) and has an output of 5 dimensions, corresponding to one of the 5 classification categories (Alum, Copper, Stainless Steel, Zinc and Brass). During model training, the model loss did not improve over multiple epochs resulting in the flat learning curves observed in Figure 11. This behavior is

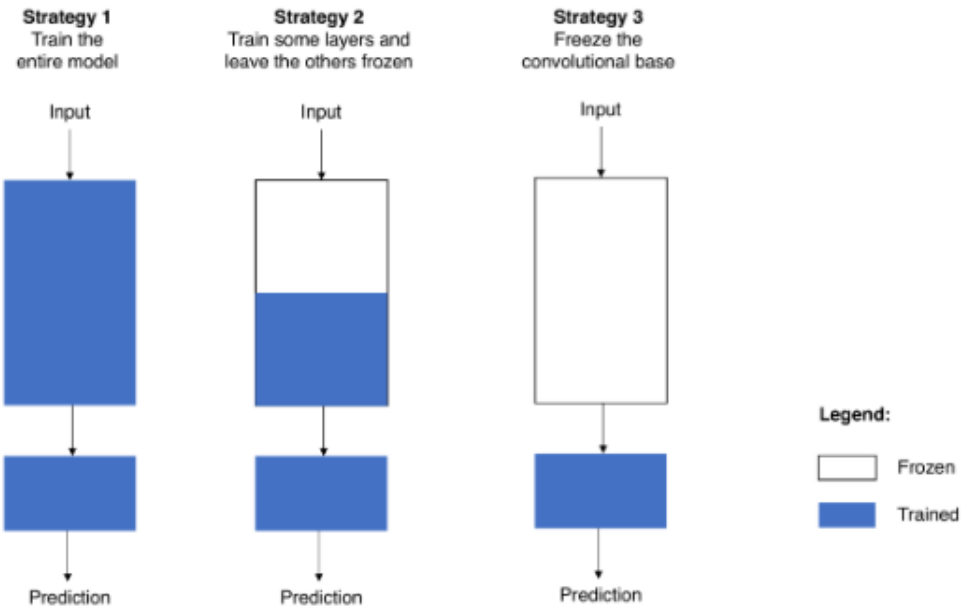

Figure 8: Strategies for training models

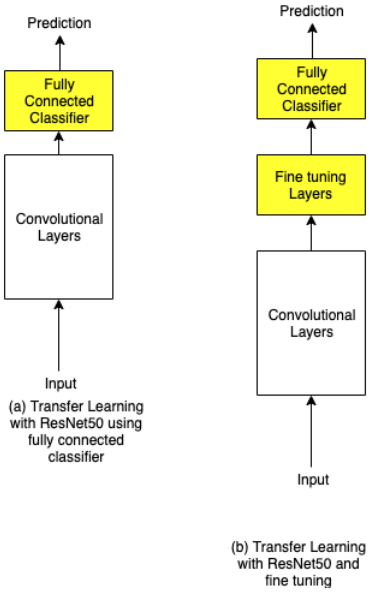

Figure 9: Transfer Learning Strategies with ResNet-50

typical when the model does not have a suitable capacity for the complexity of the dataset.

After reviewing the results of the CIFAR10 model, it was determined that a more complex model architecture was needed. A ResNet-20 model was attempted next, also using the Keras framework [7]. The smaller ResNet model was selected to reduce its size and make it suitable for edge computing. The published accuracy of the model was 92.16%.

ResNet-20

The ResNet-50 that we used for our transfer learning experiments is a deep network with 50 layers. For training from scratch we see if a less dense model can give us similar or better performance. We train on the ResNet v1 model that was used for deep residual learning of CIFAR10 Image Recognition dataset. We use a depth of 3 that results in a ResNet network with 20 layers. We compile this ResNet-20 model with limited inline image augmentation (randomly shifting images horizontally and vertically, enabling horizontal flip) to add some generalization to our model. To be consistent with our other models, we use Adam optimizer. The ResNet-20 model also gives us a validation accuracy of 0.94 and test accuracy of 0.92 which is similar to the performance with the ResNet-50 fine-tuned model. The memory footprint of the model with ResNet-20 is only 4MB which is almost 25 times smaller than that of the ResNet-50 model. Given that performance is similar, the model size is an important consideration especially if we are looking to do inference on the edge.

While our data has been carefully created

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape
=================================================================
conv2d_17 (Conv2D)           (None, 224, 224, 32)
_________________________________________________________________
activation_25 (Activation)   (None, 224, 224, 32)
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 222, 222, 32)
_________________________________________________________________
activation_26 (Activation)   (None, 222, 222, 32)
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 111, 111, 32)
_________________________________________________________________
dropout_13 (Dropout)         (None, 111, 111, 32)
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 111, 111, 64)
_________________________________________________________________
activation_27 (Activation)   (None, 111, 111, 64)
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 109, 109, 64)
_________________________________________________________________
activation_28 (Activation)   (None, 109, 109, 64)
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 54, 54, 64)
_________________________________________________________________
dropout_14 (Dropout)         (None, 54, 54, 64)
_________________________________________________________________
flatten_5 (Flatten)          (None, 186624)
_________________________________________________________________
dense_9 (Dense)              (None, 512)
_________________________________________________________________
activation_29 (Activation)   (None, 512)
_________________________________________________________________
dropout_15 (Dropout)         (None, 512)
_________________________________________________________________
dense_10 (Dense)             (None, 5)
_________________________________________________________________
activation_30 (Activation)   (None, 5)
=================================================================
Total params: 95,620,133
Trainable params: 95,620,133
Non-trainable params: 0
_________________________________________________________________
```

Figure 10: CIFAR 10 Model

in a controlled environment for keeping them consistent for training a model, we realize that in an actual metal sorting facility, this might not be the case. We therefore consider it worthwhile to add more generalization to our models. We retrain our models by adding inline augmentation to our dataset using Keras' ImageDataGenerator. Our data is augmented with random horizontal and vertical shifts, rotation in the range 0-90 degrees, zoom in the range 0.5 to 1, horizontal and vertical flips. We then evaluate our three models i.e. ResNet-50, ResNet-50 with fine tuning and ResNet-20 to see if their performance are comparable. We train all three models with a batch size of 32 for 200 epochs using Adam optimizer. Table lists the results from all three models.

We see that training from scratch with ResNet-20 gives us better performance overall with a significantly smaller model (Table 3 and Figures 13 to 18). We also notice that the ResNet-20 and

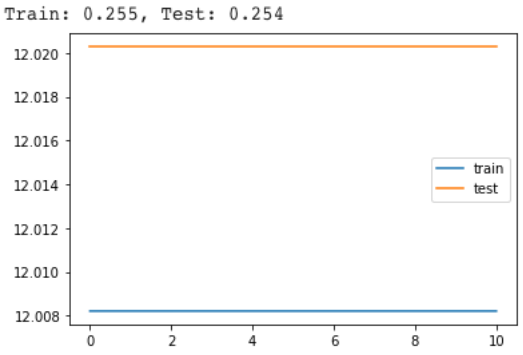

Figure 11: CIFAR 10 Test/Val

Metrics	Model 1	Model 2	Model 3
val acc	0.81	0.84	0.83
test acc	0.74	0.71	0.76
loss	0.95	1.51	0.85
size (MB)	4	90	99

Model 1 : Transfer Learning (TL) w ResNet-50
Model 2 : TL w ResNet-50 and fine tuning
Model 3 : Train from scratch w ResNet-20

Table 3: Model Performance

ResNet-50 models learn differently as evidenced by their classification reports (Figure 12). The classification reports show that ResNet-50 model is able to identify brass well but is having a hard time in classifying aluminum and stainless steel. The ResNet-20 model on the other hand, is able to classify aluminum with better but is having a tough time with brass often mistaking it for either stainless steel or zinc.

4 Results and Discussion

In addition to assessing model performance with still images, trained models were also evaluated using live video with lighting and background conditions that closely matched the environment in which the data set images were captured. Video was produced using a Logitech C920 webcam with exposure adjusted via OpenCV to encourage a uniform white background. All other camera settings were selected automatically or retained their default values. Test objects were lit using LED bulbs which resulted in rolling horizontal banding or flicker across the video frame which disrupted the uniformity of the background. Camera exposure adjustments reduced but did not completely eliminate these artifacts, so a center crop of the

camera frame was used as the input to the model to mitigate the influence of this noise. Since no negative samples were present in the training data set, an "Unknown" classification decision was created using a simple decision rule based on the precision of the model prediction: testing revealed that input video frames that were unlike training example images (e.g., a plain background with no sample material) yielded a far more precise classification prediction than video frames that contained metal samples.

From this testing it was noted that common classification errors, in approximate order of influence, appeared to be related to object color or purity, orientation, and shape. For example, brass and copper metals have distinctive colors when relatively clean and polished. However, as previously mentioned, many of the samples in the training data are dirty and rough which effectively increases the color variability within a particular object class. As a result, a sample of metal was often classified incorrectly if it was dirty.

In a similar way, highly variable sample shapes, as well as the possible orientations associated with those shapes within a video frame, presented additional challenges for the trained models. Samples of brittle metals such as zinc appear to have cracked in the metal shredding process yielding smaller samples that maintain clear evidence of their original form and function (e.g., tapped threads, sharp formed corners, and lettering serve as clear signs that zinc samples came from a metal casting). In contrast, more flexible metals such as stainless steel and aluminum exhibit very different shapes (e.g., crumples, twists, bends, etc.). However, stainless steel and aluminum appear to react to the shredding process in similar ways, and along with their similar dull gray color, it was challenging for the trained models to recognize the differences between these two classes.

In general, these variations seem to limit the evidence that might otherwise help to describe the typical features of a particular class. A larger training set size with more variations in sample purity, shape, and orientation could potentially improve model performance.

5 Conclusion

5.1 Summary

For this project, we evaluated the feasibility of using deep learning tools to classify five categories of scrap metal. Using labeled photographs of scrap metal, we succeeded in training a deep learning model with over 90% classification accuracy and to operate the model on an edge computing device with an attached webcam. Our proof-of-concept demonstrated classification accuracy that is on-par with both state-of-the-art sorting machinery and manual hand sorting.

5.2 Future Work

Given that this project originated as a collaboration with the scrap metal processing business, Alumisource, future work should focus on delivering a production-worthy sorting device. First, more realistic training data needs to be incorporated and supported. This would involve taking pictures of metal samples on a moving conveyor belt with the same lighting conditions found inside Alumisource's facility. Next, the results of the deep learning model should trigger a physical action, such as a robotic arm removing an identified, unwanted object from the moving conveyor belt. Once both of these future features are available, Alumisource would be able to realize value from this sorting device.

6 References

1. Taylor, Brian. "A World of Metal." Recycling Today, Recycling Today, 17 July 2013, https://www.recyclingtoday.com/article/rt50-metal-world-scrap-industry/.

2. "Metal Recycling Market by Metal Type (Ferrous and Non-Ferrous Metals), End-User Industry (Automotive, Packaging, Shipbuilding, Industrial Machinery) - Global Opportunity Analysis and Industry Forecast, 2014 - 2022 Update Available On-Demand." Metal Recycling Market Size, Share Trends Outlook — Forecast - 2022, https://www.alliedmarketresearch.com/metal-recycling-market.

3. Weiss, Martin. "A New Approach." Recycling Today, Recycling Today, 8 Jan. 2015, https://www.recyclingtoday.com/article/rt0115-xray-fluorescence-sorting-metals/.

4. Thung, Gary. "Garythung/Trashnet." GitHub, 10 Apr. 2017, https://github.com/garythung/trashnet.

5. "Pre-Trained Machine Learning Models vs Models Trained from Scratch." Medium, Heartbeat, 28 Oct. 2019, https://heartbeat.fritz.ai/pre-trained-machine-learning-models-vs-models-trained-from-scratch-63e079ed648f.

6. "Train a Simple Deep CNN on the CIFAR10 Small Images Dataset." CIFAR-10 CNN - Keras Documentation, https://keras.io/examples/cifar10_cnn.

7. "Trains a ResNet on the CIFAR10 Dataset." CIFAR-10 ResNet - Keras Documentation, https://keras.io/examples/cifar10_resnet/.

8. He, Kaiming, et al. "Deep Residual Learning for Image Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.90.

9. Marcelino, Pedro. "Transfer Learning from Pre-Trained Models." Medium, Towards Data Science, 23 Oct. 2018, https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751.

10. Ruiz, Pablo. "ResNets for CIFAR-10." Medium, Towards Data Science, 18 Oct. 2018, https://towardsdatascience.com/resnets-for-cifar-10-e63e900524e0.

7 Appendix

Classification Report for ResNet 50

```
Confusion Matrix
[[40  0  0  0  7]
 [ 3 17  0  2 38]
 [ 3  0 27  4  2]
 [ 0  2  0 48 11]
 [ 0  0  0  0 75]]
Classification Report
              precision    recall  f1-score   support

        Zinc       0.87      0.85      0.86        47
       Steel       0.89      0.28      0.43        60
      Copper       1.00      0.75      0.86        36
       Brass       0.89      0.79      0.83        61
    Aluminum       0.56      1.00      0.72        75
```

Classification Report for ResNet 20

```
Confusion Matrix
[[56  0  0 10  1]
 [ 2 21  0 17  0]
 [ 5  0 29  2  0]
 [17  0  0 65  0]
 [ 0  9  0  5 39]]
Classification Report
              precision    recall  f1-score   support

        Zinc       0.70      0.84      0.76        67
       Steel       0.70      0.53      0.60        40
      Copper       1.00      0.81      0.89        36
       Brass       0.66      0.79      0.72        82
    Aluminum       0.97      0.74      0.84        53
```

Figure 12: Confusion Matrix and Classification Report

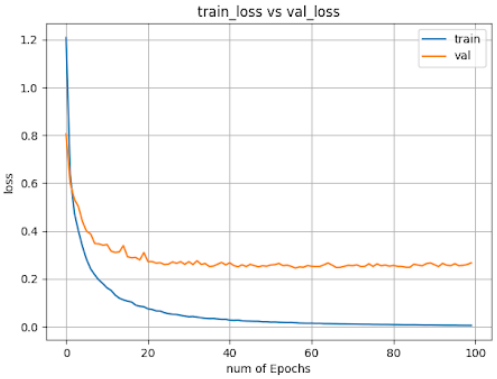

Figure 13: Train/Val. Loss (Transfer Learning - No Augmentation)

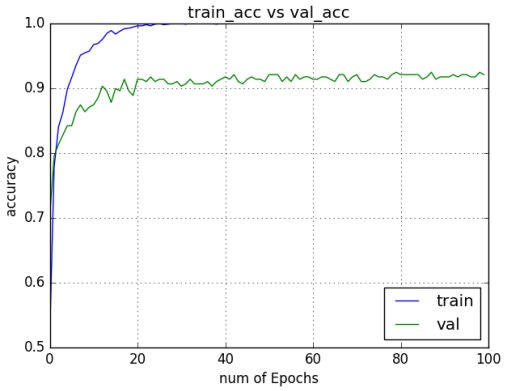

Figure 14: Train/Val. Accuracy (Transfer Learning - No Augmentation)

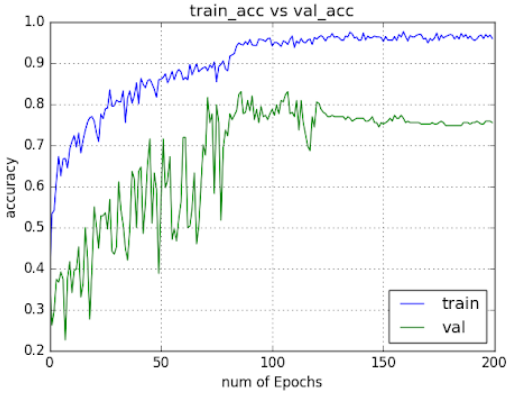

Figure 15: Train/Val. Accuracy (Resnet20 - Augmented)

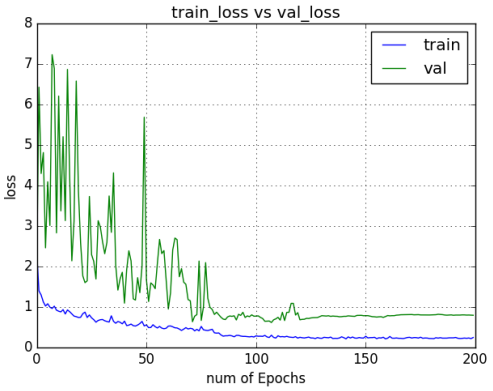

Figure 16: Train/Val. Loss (Resnet20 - Augmented)

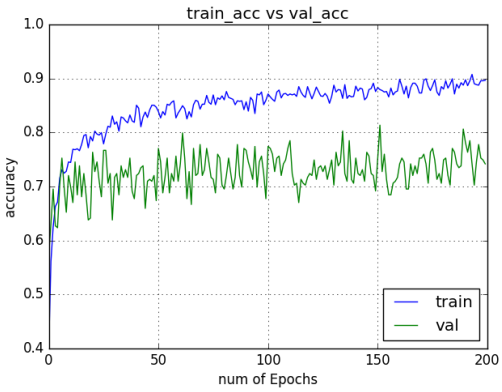

Figure 17: Train/Val. Acc (Transfer Learning - Augmented)

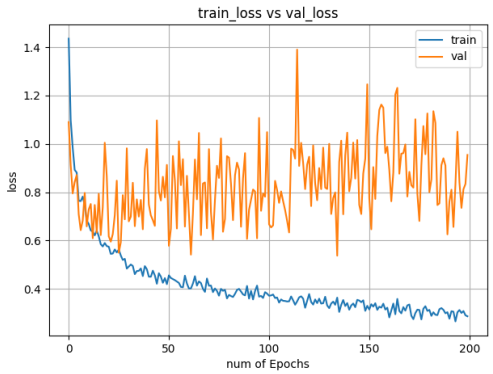

Figure 18: Train/Val. Loss (Transfer Learning - Augmented)