Image Based Machine Learning in Automotive Used Parts Identification for Remanufacturing

Abu Islam[1], Suvrat Jain[2], Nenad G. Nenadic[3], Michael G.Thurston[4],
Justin Greenberg[5], Brad Moss[6]
[1,2,3,4] Rochester Institute of Technology, Rochester, NY, 14623, USA
[5, 6] Dieselcore, Katy, TX 77449

Abstract

Remanufacturing of durable goods has the potential to prevent them from entering landfills or being melted down for recycling. Manufacturing and remanufacturing both require raw material to create a finished product. In remanufacturing this material called core, originates from previously used manufactured products. These products are rescued from landfills and scrap processors by specialized core suppliers or returned by consumers when purchasing replacement products. Due to the nature of cores being previously used or broken, identification and sorting of this material is challenging. Often core is damaged, without proper identification markings and often shows signs of rust, carbon deposits and wear. Most of the current identification, sorting and handling processes are very labor intensive, error prone and can have poor ergonomics. An automated part identification and sorting process is a potential solution for human induced error given its ability to effectively sort and inspect cores that have similar geometries or set of features. To automate the part identification, a neural network has been trained with images of different automotive parts at multiple orientations. Inception Transfer Learning has been used to reduce the number of training data requirements, speeding up training and higher accuracy. In order to keep the resolution of feature detection insensitive to parts sizes, a YOLO (You Only Look Once) algorithm has been used to create bounding box around the area of the image being analyzed. An automated sorting system has been developed that consists of a smart conveyor, multiple cameras, and laser line scanners. Once on the conveyor, the automated sorting system coordinates the movement and imaging of the core at multiple stations in order to capture a near-360 degree view of the core part. The algorithm detects the part types and models from the images captures from the vision system. It has been demonstrated that the vision based sorting system can classify parts with better than 95% accuracy. This paper will present the results in detail.

Introduction

Remanufacturing of durable goods has the potential to prevent them from entering landfills or being melted down for recycling. Manufacturing and remanufacturing both require raw material to create a finished product. In remanufacturing, this material called core, originates from previously used manufactured products. These products are rescued from landfills and scrap processors by specialized core suppliers or returned by consumers when purchasing replacement products. Due to the nature of cores being previously used or broken, identification and sorting of this material is challenging. Often core is damaged, without proper identification markings and often shows signs of rust, carbon deposits and wear. Most of the current identification, sorting and handling processes are very labor intensive, error prone and can have poor ergonomics. An automated part identification and sorting process is a potential solution for human induced error given its ability to effectively sort and inspect cores that have similar geometries or set of features.

A remanufacturing facility is a type of manufacturing center that uses previously used or broken finished goods to manufacture sellable finished product that meets or exceeds the original product manufacturing specifications. Each previously used finished product is received into the remanufacturing center, properly identified, and sorted. These products are then completely disassembled, reusable components are compared to original product specifications and consumable components are discarded and replaced with new. Reusable components that do not meet original product specifications are machined to specification or replaced. The product is then assembled and tested to validate meeting original specifications before being packaged and returned to the parts distributors for sale. One of the major problems in this industry is the accuracy of identifying core material and sorting them for future consumption. The identification of core historically has relied on years of experience and intrinsic industry knowledge. This results in labor shortages of employees with the knowledge, skills, and abilities to do the job. If a part is wrongly identified, it could lead to a delay or the complete halt of remanufacturing. For this problem, an object detection and recognition system can be used to automatically identify the different parts and put them into the required category.

Literature Review

Though there are multiple approaches available for solving this problem, we have chosen deep learning as it works well with large volume of data and scales up well to support real time identification. Due to large volume of data, training models such as convolutional neural networks (CNN) is ideal as they can easily accommodate huge amount of samples while providing accuracy in predictions. There are several approaches proposed by researchers in the field of object detection and recognition.

Although there have been numerous advancements in the field of object detection, challenges still persist. (Lowe, 2004) proposed SIFT (Scale Invariant Feature Transform) algorithm which was a promising start but it is known to be computationally expensive and at times inaccurate. This resulted in (Bay et al., 2008) proposing SURF (Speeded Up Robust Features) which improved upon SIFT algorithm by reducing the computational cost but its performance wasn't satisfactory enough to be applicable in a use case like ours dealing with production environment, such as a remanufacturing facility. While various deep learning models such as convolutional neural networks (CNNs) and deep neural networks (DNNs) have been applied to train better object detection systems, the computational cost and speed is still an issue. (Kim et al., 2020; Rucco et al., 2019) proposed deep learning approaches of mechanical parts identification, but all the parts data were from CAD images. (Borrelly & Laurgeau, 1980) proposed a means to identify mechanical components in a moving conveyor. Most existing methods (Yanagisawa et al., 2018),(Erhan et al., 2014) are not able to handle real-time constraints for applications like automotive parts detection in remanufacturing that need high-level classification and localization accuracy. In our use case, remanufacturing industry consists of a lot of complications and difficulties like varying lighting conditions, background distortion, and part orientation which calls for a more robust, efficient, and fast object detection approach. From a lot of previous research, it has been found that object detection systems can be implemented in different ways (Zou et al., 2019). A lot of them have been developed and tested on several datasets. However, the scope and feasibility of object detection systems for automotive parts in the remanufacturing industry is not yet explored completely. Therefore, in this paper we are proposing an approach using the YOLO (You Only Look Once) algorithm (Ahmad et al., 2020; Bochkovskiy et al., 2020) for real-time object detection and recognition. Lightweight and fast, YOLO reduces the complexity of deep neural networks while still maintaining high accuracy. By allowing only one forward pass through the network, it is possible to build an object detector that is capable of processing images in real time. Previous algorithms required many passes through the image to detect objects, which resulted in reduced efficiency. The main motivation behind this project is to have an efficient real time solution to detect and classify objects which are used in the remanufacturing industry like automotive parts.

Goals

Figure 1. Brake Shoe Cores

Figure 2. Fuel Injector Cores

The goal of the project is to first sort between different types of core, brake shoes vs fuel injectors. Then sort between make and models within each type of component. It is also necessary to recognize core with accuracy of better than 90% within a cycle time of 10 seconds or less. Automation for identification recognition of components makes the system as robust as possible, flagging any new parts that the system has not been trained on.

Experiment Description

Figure 3. Test Fixture

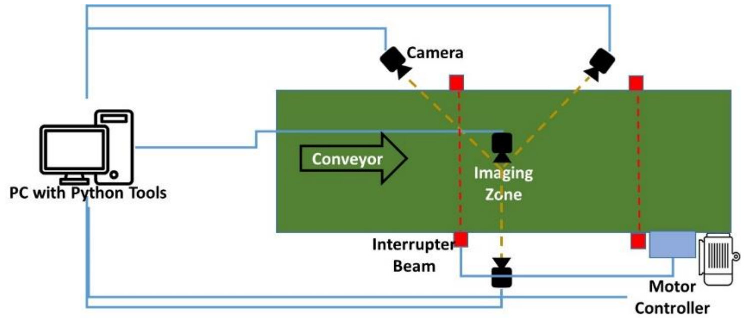

Figure 4. Fixture Schematic

An automated sorting system has been developed that consists of a smart conveyor with multiple cameras as shown in Figure 3. Dorner 3200 series conveyor 120" L x 18" W was chosen for its compact footprint. The belt material was chosen to minimize glare and reflections off the belt. The belt is driven by variable speed .75 HP VFD gear motor capable of maintaining full torque throughout belt speed range of 6.9 - 69 ft/min. The motor is controlled by Lenze AC Tech Protech i550 Inverter. The belt also has 2 Photo Eye Kit, 24V DC retro reflective sensor, one for detecting parts at the imaging zone and the other to detect end of travel. Imaging Source cameras with Sony CMOS IMX264LQR, 5MP, 38 fps image sensors were chosen for image capture. Cameras were mounted with Kowa 1" 16.0mm F1.4 Manual Iris C-Mount Lens. Edmunds Optics M35.5X0.5 Linear Glass Polarizer filters were used to minimize ambient light disturbances. Smith-Victor Slim Panel 800W 2-Light Daylight LED Kit was used for illumination. Light diffuser boards placed on the top of the fixture and lights off the illuminator was made to reflect off the diffuser boards to minimize glare. The cameras are connected to the computer via USB cable. The reflective sensors are connected to the PLC digital input ports. The PLC communicates with the PC thru Ethernet cable as shown in Figure 4. The Control software in written in Python ecosystem. Once on the conveyor the part moves towards the imaging zone. Once the optical sensor detects the part, the cameras capture images and the part moves on. The image is sent to the control computer thru USB. The algorithm in the control PC detects the part types and models from the images captured by the vision system.

Model Used

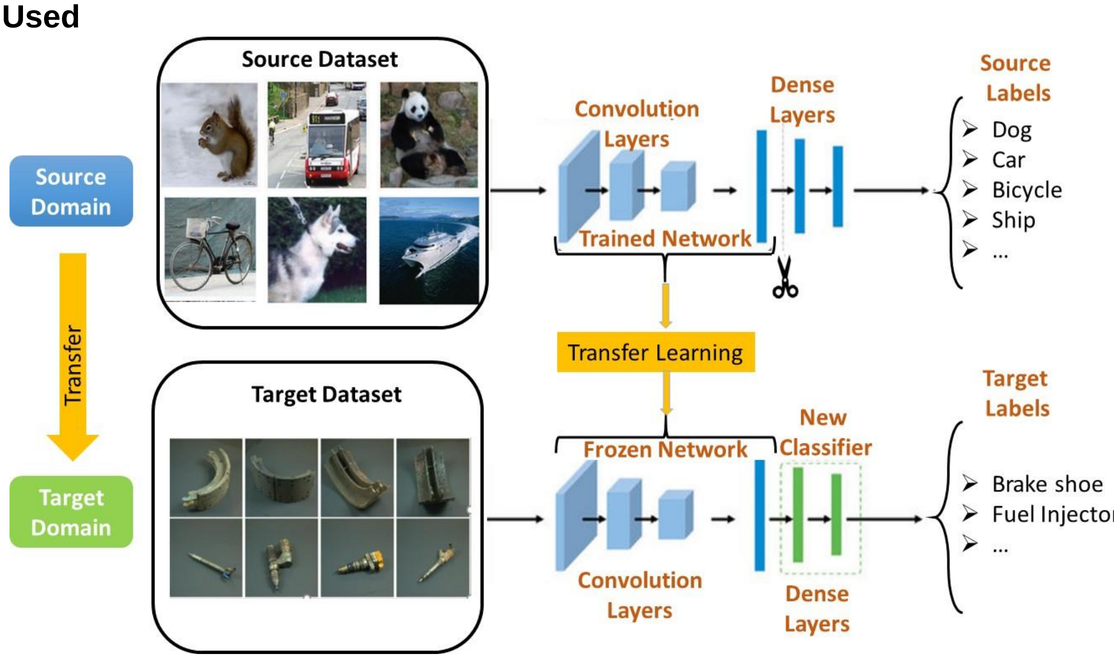

Figure 5 Inception Transfer Learning Model

To automate the core identification, a neural network has been trained with images of different core at multiple orientations. Inception Transfer Learning as shown in *Figure 5*, has been used to reduce the number of training data requirement, speeding up training and higher accuracy.

Type	Part No	OEM	description	Number
Fuel Injectors	445117016	Bosch	6.7L Ford injector	6
	AE	Ford	7.3L Ford injector	5
	OR4528	CAT	3176 injector	5
	445120082	Bosch	LMM injector	5
	445120027	Bosch	LLY injector	5
Brake Shoes	4707Q	Meritor	Drum Brake Shoe	2
	4718Q	Meritor	Drum Brake Shoe	4
	4725E	Eaton	Drum Brake Shoe	4
	4524Q	Meritor	Drum Brake Shoe	4
	4729E	Eaton	Drum Brake Shoe	4
	4515Q	Meritor	Drum Brake Shoe	3

Figure 6. Training Data

Figure 7. Labeled Brake shoe

Figure 8. Labeled Fuel Injectors

As shown in Figure 6 the model was trained using 5 different fuel injectors and 6 different brake shoe models supplied by DieselCore. Each part model had multiple replications as shown in Figure 7 and Figure 8. Multiple pictures were captured from each part and their labels used for ground truth generation.

Results

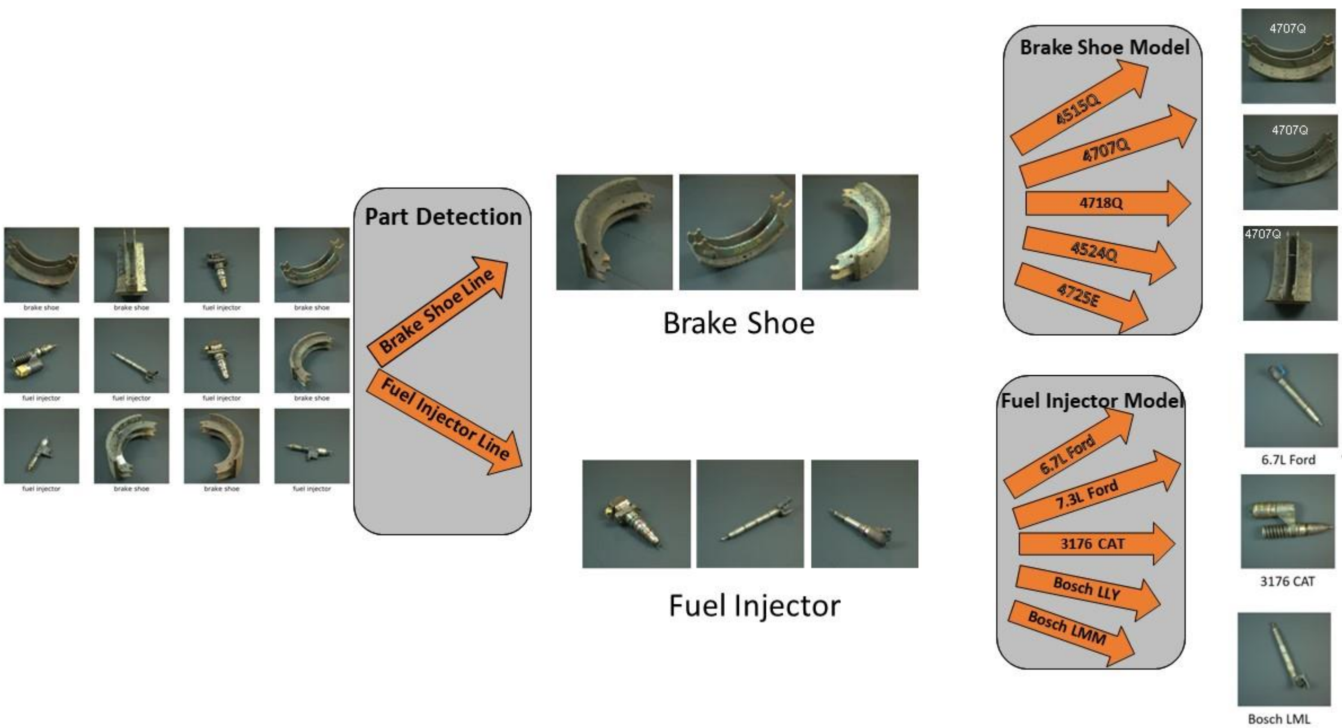

Figure 9. Process Flow

As the cameras capture new images from the parts in the conveyor, the images are first classified using Brake shoe-Fuel Injector model. Once the type of part is determined, the images are further classified using Brake shoe or Fuel Injector model classifier as shown in Figure 9.

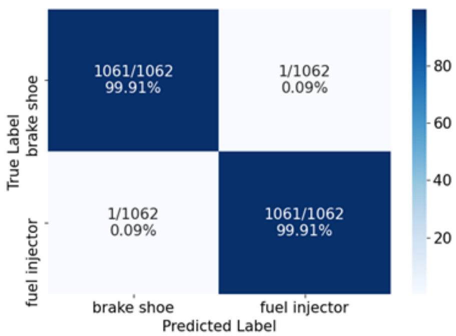

Figure 10. Brake Shoe Vs Fuel Injector Results

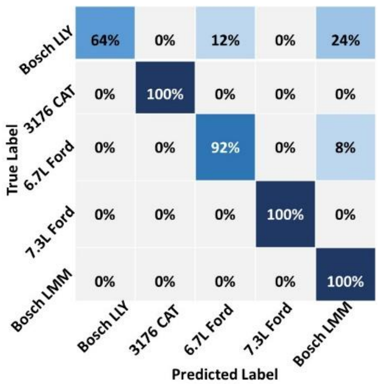

Figure 11. Fuel Injector Results

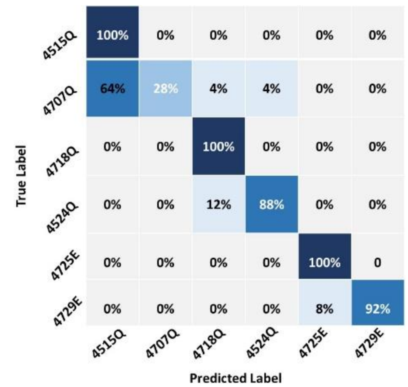

Figure 12. Brake Shoe Results

As can be seen from Figure 10, the model is able to distinguish between a fuel injector and a brake shoe with a high degree of accuracy. When it comes to distinguishing between different types of fuel injectors, the model works quite well except for the case of Bosch LLM+LLY vs Bosch LML as showing in Figure 11. Similarly, the model works quite well for identifying different models of brake show, except for 4515Q and 4707Q as shown in Figure 12. Upon further investigation about causes for these errors, it was found that the reason for the fuel injector errors was, the fuel injector sizes are small in the field of view of the camera, so the feature details are not clean in the region of interests as shown in Figure 13. Similarly, as can be seen from Figure 14, the main difference between the two models are the distance between the 2 rivet holes which could be difficult to capture in the images. Figure 14

Figure 13. Different Part Size Issues

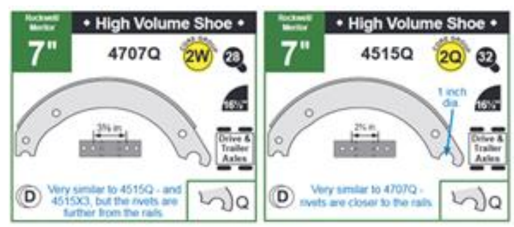

Figure 14. Hidden Features Issue

In order to keep the resolution of feature detection insensitive to parts sizes, a YOLO (You Only Look Once) algorithm has been used to create a bounding box around the area of the image being analyzed. YOLO treats object detection as a regression problem to predict what objects are present and where they are.

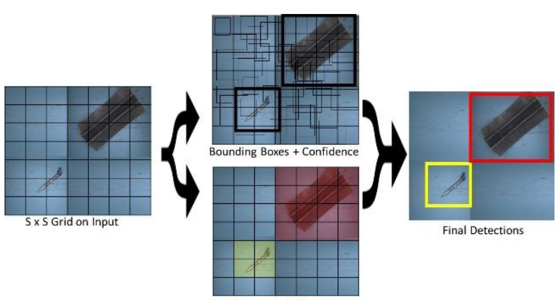

Figure 15. YOLO Grid Architecture

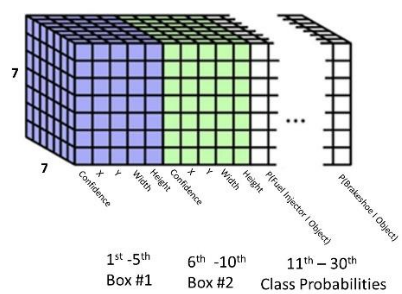

Figure 16.YOLO Data Strucutre

YOLO uses a single neural network that predicts class probabilities and bounding box coordinates from an entire image in one pass. The YOLO model divides the image into S x S grid, shown in Figure 15. Each of these grid cells predicts B bounding boxes and confidence scores for these boxes. If the center of an object falls into one of the grids, then that cell is responsible for detecting that object. Each bounding box outputs five predictions: the center coordinates, the length and width of the bounding box and the class prediction as showing in Figure 16.

The loss function defined in YOLO as follows

$$Loss_{yolo} = \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\right]+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]+ \longrightarrow \text{Bounding Box coord}$$

$$\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(C_i-\hat{C}_i\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}\left[\left(C_i-\hat{C}_i\right)^2+ \longrightarrow \text{Confidence}\right.\right.$$

$$\sum_{i=0}^{S^2}1_{ij}^{noobj}\sum_{C\in classes}\left[\left(p_i(C)-\hat{p}_i(C)\right)^2\right] \longrightarrow \text{Classification}$$

where,

- x_i, y_i denote the location of the centroid of the anchor box
- w_i, h_i, denote the width and height of the anchor box
- C_i, denote is the confidence score of whether there is an object or not, and
- $p_i(c)$ denote the classification loss.
- $\mathbf{1}_i^{obj}$ denotes if object is present in cell ***i***.
- $\mathbf{1}_{ij}^{obj}$ denotes j_{th} bounding box responsible for prediction of object in the cell ***i***.
- $\lambda_{coord} = 5$ and $\lambda_{noobj} = .5$ are regularization parameter required to balance the loss function
- All losses are *mean-squared* errors, except classification loss, which uses *cross-entropy* function.

The confidence score indicates how sure the model is that the box contains an object and also how accurate it thinks the box is that predicts. The confidence score can be calculated using the formula:

confidence_score = Pr(object) * IoU

IoU: Intersection over Union between the predicted box and the ground truth.
If no object exists in a cell, its confidence score should be zero.Each bounding box consists of five predictions: **x, y, w, h**, and confidence.Each grid cell also predicts C conditional class probabilities ***Pr(Classi|Object)***. It only predicts one set of class probabilities per grid cell, regardless of the number of boxes B. During testing, these conditional class probabilities are multiplied by individual box confidence predictions which give class-specific confidence scores for each box. These scores show both the probability of that class and how well the box fits the object.

Pr(Class i|Object)*Pr(Object)*IoU = Pr(Class i)*IoU.

The final predictions are encoded as an S x S x (B*5 + C) tensor.

The base model has 24 convolutional layers followed by 2 fully connected layers. It uses 1 x 1 reduction layers followed by a 3 x 3 convolutional layer. Fast YOLO uses a neural network with 9 convolutional layers and fewer filters in those layers. The final layer uses a linear activation function. The rest uses a leaky ReLU. The complete network is shown in the figure.

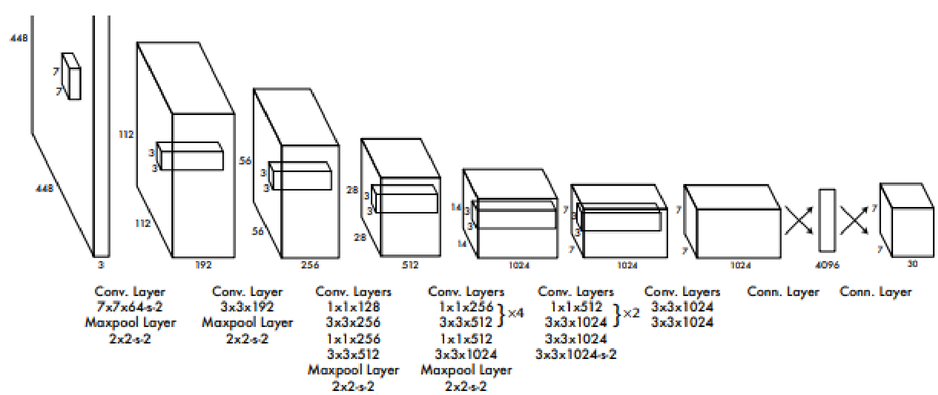

Figure 17 YOLO Network Architecture

Network Architecture

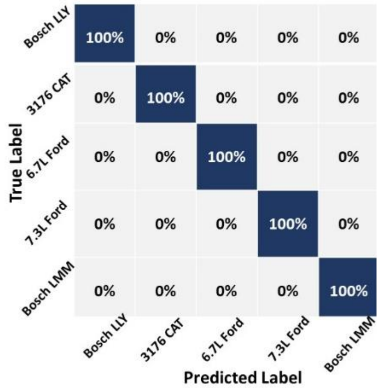

Figure 18. Fuel Injector Detection

Figure 19. Fuel Injector Results

Figure 20. Brake Shoe Detection

As can be seen from Figure 18 that YOLO algorithm can not only detect, localize with bounding box, and classify small objects in the field of view, but it can also detect multiple objects successfully simultaneously. Figure 19 shows the confusion matrix for Fuel injector classification and it completely eliminated the errors between Bosch LML and Bosch LML+LLY parts. Figure 20 shows corresponding results for Brake shoe, as the Brake shoes are large compared to the field of view, it didn't show any improvement in accuracy from previous approach.

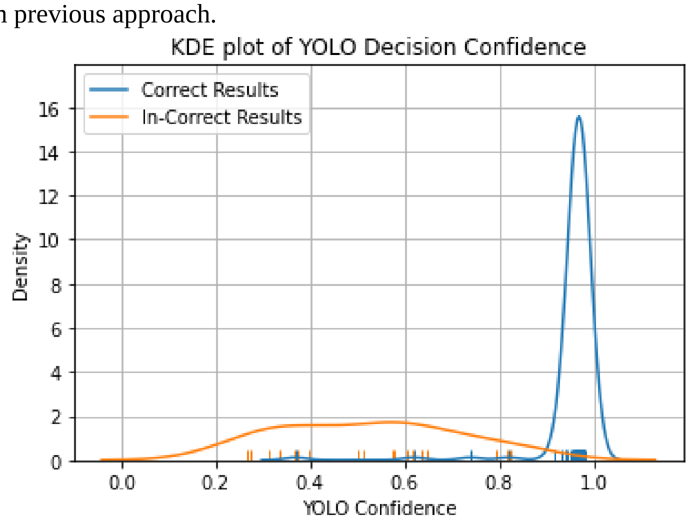

Figure 21 Confidence of YOLO decision

Figure 21 shows the Confidence of YOLO decision. Although the confidence associated with most correct decisions (blue curve) are quite high (greater than 90%) there are some correct decisions with confidence as low as 40%. The model was tested with parts it is not trained on to test the failure modes under extreme conditions. It can be seen there are some incorrect decisions with high confidence with confidence greater than 90%. This warrants a more complex probabilistic decision making than choosing a single confidence threshold to know that the predictions are correct.

Combining classifiers

In order to improve the robustness of detection accuracy, images from multiple cameras can be utilized. Images from four different directions (corresponding to the different camera orientations) provided data for training four separate classifiers. The four classifiers provide independent probability estimates $\hat{p}_k$.

The k^{th} output of a classifier with K outputs can be viewed as the estimated probability (Goodfellow et al., 2016)

$$\hat{p}_{\mathrm{k}} = \mathrm{softmax}(z_k) = \frac{\mathrm{e}^{z_k}}{\sum_{i=1}^{K} \mathrm{e}^{z_i}} \qquad (1)$$

Where, the condition $\sum_{k=1}^{K} \hat{p}_k = 1$ is automatically satisfied. YOLO also provides confidence associated with its predictions (Redmon et al., 2016)

Because the probability estimates can be mapped readily onto a multinomial distribution, they can be efficiently fused. The multinomial distribution is given by (Bishop, 2006)

$$p_{Mult[k]} = \prod_{i=1}^{K} \hat{p}_k^{\delta_{i,k}}, \qquad k \in \{1,2,\dots,K\} \tag{2}$$

where the $\delta_{i,k}$ is the Kronecker function, which is 1 if $i = k$ and 0 if $i \neq k$.

The multinomial distributions are probabilistically combined in the Bayesian sense using the Dirichlet distribution. This approach is consistent with coherent reasoning (Koller & Friedman, 2009; Pearl, 2014) and is viewed as an extension to logic in physical sciences (Jaynes et al., 2003).The Dirichlet distribution is the conjugate prior to the multinomial distribution given by

$$p_{Dirichlet}(x_1, x_2, \dots, x_K | \alpha_1, \alpha_2, \dots, \alpha_K) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} x_k^{\alpha_k - 1}, \quad 0 \leq x_k \leq 1 \ \forall k \text{ and } \sum_{k=1}^{N} x_k = 1 \tag{3}$$

And $\Gamma(x)$ is the gamma function

$$\Gamma(x) = \int_0^{\infty} t^{x-1} \mathrm{e}^{-t} dt, \quad x > 0 \tag{4}$$

The updates using the conjugate priors are straightforward and fast (in contrast to the general Bayesian updating, which are involved and require computationally expensive evaluation of integrals).

In some cases, multiple estimated probabilities have comparable values. When these multiples include the maximum, the confidence of the decision is relatively low. Classifiers often employ the reject option to avoid making low confidence decisions by setting the minimum level of the estimated probability $\theta_{\min}$ (Bishop, 2006)

$$\max(\hat{p}_k) < \theta_{\min} \tag{5}$$

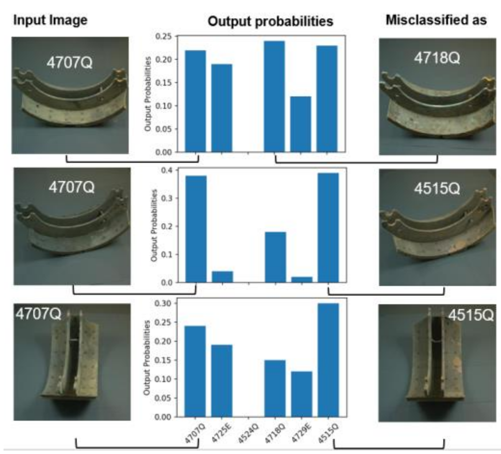

Figure 22. Difficulty with similar parts

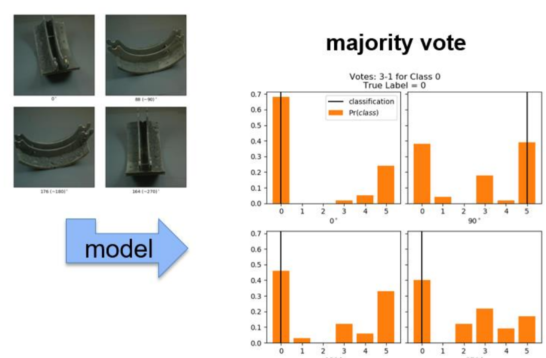

Figure 23. Voting for better classification

If the condition given by Eq. (1) is not satisfied, the automated system rejects to make the decision and involves the human into the decision loop. If $\theta_{\min}$ is set to 1, the system will never make the decision; if it is set to $1/K$ it will never reject to make the decision. The user and designer set a sensible threshold based on the track record of the performance.

In addition to setting the minimum value for the maximum estimate, we can specify the minimum difference between the two highest estimates

$$\max(\hat{p}_k) - \max_{k \neq \operatorname{argmax}(\hat{p}_k)}(\hat{p}_k) < (\Delta\theta)_{\min} \tag{6}$$

Figures 22 and 23 are important in demonstrating the challenges faced in identifying similar looking objects in computer vision and how the use of multiple cameras can help overcome these challenges. In Figure 22, we can see examples of two brake shoe parts, 4707Q and 4718Q, whose output probabilities are too similar to make a decision as to which part it actually is. This is a common problem in computer vision where similar looking objects can lead to confusion in classification. To address this issue,

Figure 23 demonstrates the use of a probabilistic voting system based on multiple cameras to aid in better decision making. In this approach, multiple cameras capture images of the same part from different angles, and the detections are aggregated to make a final decision. If a given part's probability or confidence of detection is higher in majority of the detections with different cameras, we can conclude that the part majorly detected is the actual part in the image. The significance of these figures lies in their ability to showcase the limitations of single-camera detection and the potential benefits of using multiple cameras in computer vision applications. By demonstrating how a probabilistic voting system can improve detection accuracy, these figures can inform the development of more robust and reliable computer vision systems

Conclusions and Recommendations

The remanufacturing of used parts can be a viable alternative to purchasing new parts. Like manufacturing, the process begins with correctly identified raw materials for production. Due to the source of acquisition of this raw material known as core, an accurate and efficient identification process is essential. This process can be overseen manually, but an automated system has the potential to provide more accurate, consistent, and efficient results. This paper presents the implementation of an automated visual recognition system for automating the identification and sorting of various core parts. The system uses computer vision and object detection techniques to identify and separate different types of core based on their characteristics. The algorithm detects the part type and model from the image. The part is then identified and sorted accordingly. This process effectively provides identification and sorting of material with greater than 95% accuracy within the constraints of 10 seconds or less cycle time. This solution is reproducible and new products can be added to the learning model as requirements emerge. Proper implementation of this solution reduces reliance on manual error prone human candidates with the knowledge, skills, and abilities to complete the task. As a future work more rigorous statistical decision making based on the theoretical approach presented in this paper will be conducted.

Acknowledgements

Funding for this work was provided by the New York State Department of Economic Development under Grant #AC118. Any opinions, results, findings and/or interpretations of data contained herein are the responsibility of the authors and do not necessarily represent the opinions, interpretation or policy of New York State.

References

Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., Haq, A. U., & Ali, R. (2020). Object Detection through Modified YOLO Neural Network. *Scientific Programming*, *2020*, 1–10. https://doi.org/10.1155/2020/8403262

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359. https://doi.org/10.1016/J.CVIU.2007.09.014

Bishop, C. M. (2006). Pattern recognition and machine learning. In *Information science and statistics.* Springer.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. http://arxiv.org/abs/2004.10934

Borrelly, J. J., & Laurgeau, C. (1980). *Recognition of Mechanical Parts on a Moving Conveyor.* 577–581. https://doi.org/10.1016/s1474-6670(17)64763-3

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2155–2162. https://doi.org/10.1109/CVPR.2014.276

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Jaynes, E. T., Bretthorst, G. L., & Inc., ebrary. (2003). Probability theory the logic of science. In *ebrary Electronic Books.* (pp. xxix, 727 p.). Cambridge University Press.

Kim, S., Chi, H. gun, Hu, X., Huang, Q., & Ramani, K. (2020). A Large-Scale Annotated Mechanical Components Benchmark for Classification and Retrieval Tasks with Deep Neural Networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12363 LNCS*, 175–191. https://doi.org/10.1007/978-3-030-58523-5_11

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision* (Vol. 60, Issue 2).

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December,* 779–788. https://doi.org/10.1109/CVPR.2016.91

Rucco, M., Giannini, F., Lupinetti, K., & Monti, M. (2019). A methodology for part classification with supervised machine learning. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, *33*(1), 100–113. https://doi.org/10.1017/S0890060418000197

Yanagisawa, H., Yamashita, T., & Watanabe, H. (2018). A study on object detection method from manga images using CNN. *2018 International Workshop on Advanced Image Technology, IWAIT 2018*, 1–4. https://doi.org/10.1109/IWAIT.2018.8369633

Zhou, L., Garg, A., Zheng, J., Gao, L., & Oh, K. (2021). Battery pack recycling challenges for the year 2030: Recommended solutions based on intelligent robotics for safe and efficient disassembly, residual energy detection, and secondary utilization. *Energy Storage, 3*(3). https://doi.org/10.1002/est2.190

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). *Object Detection in 20 Years: A Survey*. 1–39. http://arxiv.org/abs/1905.05055

About the Authors

Abu Islam received his Ph.D. in Mechanical Engineering from Rensselaer Polytechnic Institute (Troy, NY, USA) in 1995. He is the Lead Staff Scientist of Golisano Institute of Sustainability of RIT. His research interests include the use of computer vision, data analytics and artificial intelligence in remanufacturing and health monitoring of machines. He holds a Black Belt in Design for Lean Six Sigma and 26 US Patents.

Michael G. Thurston received his Ph.D. in Mechanical and Aerospace Engineering from the University of Buffalo (Buffalo, NY,USA) in 1998. He is the Technical Director and Research Associate Professor at the Center of Integrated Manufacturing Studies at Rochester Institute of Technology. He formerly held positions in air conditioning system development at General Motor and Delphi, and as a Researcher at the Applied Research Laboratory at Penn State University. He holds 7 patents in the areas of air conditioning and asset health monitoring.

Nenad G. Nenadic received his Ph.D. in Electrical and Computer Engineering from University of Rochester (Rochester, NY, USA) in 2001. He is a Research Associate Professor of Center for Integrated Manufacturing Studies (CIMS) at Rochester Institute of Technology. His research interest include design, analysis, and monitoring of electromechanical devices and systems. He has two patents in electromechanical design. He co-authored a textbook *Electromechanics and MEMS* and is a member of IEEE.

Suvrat Jain is a Graduate Student of Software Engineering Department of Rochester Institute of Technology, specializing in Data Science. He also holds a Bachelor of Technology in Electronics and Communucations from Manipal Academy of Higher Education, Dubai. He also has professional experience at Merkle doing cloud based salesforce marketing.

Justin Greenburg is the President and CEO of DieselCore. He has consulted with multiple OE, Tier 1 and aftermarket remanufacturing companies. Justin has served as Chairman of the Board of Directors for MERA, The Association for Sustainable Manufacturing. He also holds a Board of Director position with MEMA, the Motor and Equipment Manufacturing Association. He is the recipient of the 2015 Houston Business Journal Fast 100 Award, also 2015 2016 Inc 5000 Fastest Growing Companies in America Awards.

Brad Moss is the Director of Technology, Research and Development of DieselCore. He received his B.S. in Business Administration from Western Governs University. He is the Manager of technology and development. He oversees and participates in developing solutions to business requirements.