Position Paper: Reducing Amazon's packaging waste using multimodal deep learning

Prasanth Meiyappan
Amazon.com
meiyappa@amazon.com

Matthew Bales
Amazon.com
mjjbales@amazon.com

ABSTRACT

Since 2015, Amazon has reduced the weight of its outbound packaging by 36%, eliminating over 1,000,000 tons of packaging material worldwide, or the equivalent of over 2 billion shipping boxes, thereby reducing carbon footprint throughout its fulfillment supply chain. In this position paper, we share insights on using deep learning to identify the optimal packaging type best suited to ship each item in a diverse product catalog at scale so that it arrives undamaged, delights customers, and reduces packaging waste. Incorporating multimodal data on products including product images and class imbalance handling technique are important to improving model performance.

1. INTRODUCTION

Amazon is committed to reaching net zero carbon by 2040, a decade earlier than the goals of the Paris Agreement [1, 2, 3]. Amazon sells hundreds of millions of different products and ships billions of items a year. Since 2015, Amazon has reduced the weight of its outbound packaging by 36%, eliminating over 1,000,000 tons of packaging material worldwide, or the equivalent of over 2 billion shipping boxes thereby reducing carbon footprint throughout its fulfillment supply chain [1, 2, 3]. The reduction is in large part due to the use of machine learning to identify the optimal packaging type for each product so that it arrives undamaged, delights customers, and reduces packaging waste [1, 2, 3]. The following examples of available package types are ranked from most protective to least protective (see Figure 1): corrugate boxes, padded mailers or envelopes, poly or paper bags, and frustration free packaging (FFP) or ship in own container (SIOC) which have only vendor provided packaging.

In this position paper, we highlight learnings from using deep learning [4] to identify products that are suited for a given package type. The novelty of this article is two-fold: First, it contributes to sparse public literature on using machine learning to select outbound shipment packaging [5, 6]. Existing packaging literature are limited to either machine learning on small datasets (less than one million products) or using only textual/tabular information on products. In contrast, our learnings are based on a larger dataset (several million products) and use multimodal information on products, including product images. Second, the dataset in packaging domain typically has class imbalance (minority group is 1%-8% of sample data). There is little research on evaluating deep learning in the context of both class imbalance and multimodal data [7]. We highlight learnings from using different data-level and algorithm-level techniques to handle class imbalance in the packaging domain. The broader idea of using multimodal data including product images to assess package suitability has been filed for patent [8] and we limit details to business non-critical and non-confidential information.

2. THE PACKAGING DOMAIN

2.1 Problem Statement

The problem we faced was how to create a scalable mechanism to identify optimal packaging across the hundreds of millions of products shipped by Amazon, using the information provided by vendors and Amazon systems (e.g., warehouse technology that captures product images). An example to contextualize the waste impact of pack type selection: a padded paper mailer is 75% lighter and occupies 40% less space when shipped compared to a similar size box [1]. However, we balance the trade-off between packaging waste considerations versus the likelihood that a customer receives a damaged item because of less protective/wasteful packaging. For example, fragile products will necessitate more protective packaging such as a box, while sturdier products could ship in flexible package types or in FFP/SIOC (Figure 1).

We pose optimal package selection as a binary classification problem where the goal is to classify products suitable for a given package type. We collect labeled training data through various channels including manual labeling of products by trained labelers and through direct customer feedback signals. We typically have several million training labels for any given pack type and the data has extrinsic class imbalance (typically 1% to 8%) where the minority group are products that are unsuitable for a given pack type. We use customer facing product features that include textual data such as product title and product description, numerical data such as product weight and dimensions, and categorical data such as product assignments to catalog product categories. We also incorporate images on how the product is packaged by the vendor (Figure 2). Product images are crucial input features overlooked in earlier studies [e.g. Refs 5, 6]. For example, a machine learning model that solely relies on textual features may predict a LED light bulb would require box packaging; however, the product image may actually indicate the vendor already packed it safely in a box thereby making it suitable for SIOC which has only the vendor provided packaging (Figure 2).

Figure 1: Example outbound packing types at Amazon.

Main product Image on the website	Product as given by the vendor	Description
		Product is an LED Light Bulb. We ship it as SIOC i.e., ship in vendor provided packaging with no additional packaging as the image indicates the product is already packaged in a box.
		Product is a floor mat. We ship using a bag as its optimal level of protection needed to arrive undamaged.
		Product is a soccer ball. We ship using a bag because the image indicates it's deflated. An inflated ball on the other hand would require a box because it could roll on conveyor belt and in trucks if shipped in a bag.

Figure 2: Some examples of product images.

2.2 Performance Metric and Multimodal Deep Learning

As our data is highly skewed, we use Precision-Recall (PR) curve as our primary performance metric following Ref. [9]. We also use a set of complementary performance metrics (e.g., Brier score) to capture different aspects of model performance [9]. For simplicity, we limit our discussion to PR curve.

We use a multimodal deep model that learns from both product images and textual/tabular descriptions of the products. We leverage existing deep learning backbones to extract features from each modality (Figure 3). We use a fuse-late strategy [10] to combine feature representations from each modality near the output layer. The technique allows the model to extract higher level features from each modality before deciding how to fuse them. We pre-process the product images using faster R-CNN [11] to crop an image to the product area. To extract image features, we fine-tuned a ResNet-50 architecture [12] that was pre-trained on ImageNet data [13]. Given the diverse nature of the product catalog, the product textual data (e.g., product descriptions contributed by sellers) may have out-of-vocabulary words such as jargon and abbreviations. Therefore, for textual features, we used FastText [14] character-based word embeddings and use weighted averaging of word embeddings to derive product level textual features. For tabular data (numerical and categorical), we use one hidden-layer Multilayer-Perceptron to generate vector representations. We concate-

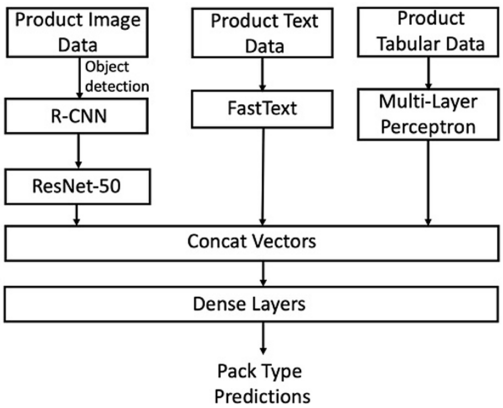

Figure 3: Multimodal deep model that leverages existing model backbones. We train the model end-to-end using images, text, and tabular (numerical and categorical) data.

Table 1: Summary of imbalance handling methods explored with observed change in model performance. We measure model performance using PR curve on the holdout set with original skewed distribution. We report performance as % change in PR-AUC when compared to the multimodality baseline where class imbalance is not addressed. Range provided is across different pack types.

Technique	**Technique Type**	**Relative PR-AUC Change**
Borderline SMOTE Over Sampling	Data	4% to 7%
Near-miss Under Sampling	Data	-7% to 0%
Hybrid of Random Over Sampling and Random Under Sampling	Data	6% to 10%
Two-phase Learning with Random Under Sampling	Data	18% to 24%
Mean Squared False Error	Algorithm	9% to 14%
Focal Loss	Algorithm	2% to 5%

nate the topmost vectors from each modality using two dense layers to produce pack type predictions. Overall, using multimodal data helped improve model performance (PR curve) on pack type classification by as much as 30% across different pack types compared to corresponding single modality baselines. We observed that the multimodal deep model performance is most sensitive to how we handle class imbalance compared to other model hyper-parameters as well as alternative deep learning backbones we tested.

2.3 Handling Class Imbalance

We summarize learnings from six different techniques to handle class imbalance (see Table 1). Four are data techniques and two are algorithm techniques. There are many more approaches to handle class imbalance including hybrid approaches [7]. The techniques we discuss were both practical to implement and satisfied our time and cost constraints for long-term maintenance. For all techniques, we trained the model to adequate epochs to minimize performance differences that may arise from slow convergence [15].

Data techniques In data technique, we modified the training data to reduce class imbalance. We tested four approaches: *Borderline SMOTE Over Sampling* [16], *Near-miss Under Sampling* [17], *Hybrid of Random Over Sampling and Random Under Sampling*, and *Two-phase Learning with Random Under Sampling* [18]. In *Borderline SMOTE Over Sampling*, we duplicated minority samples to achieve class balance using *Borderline SMOTE* [16]. This approach resulted in PR-AUC increase by 4%-7% across pack types. The biggest disadvantage is the training took 25%-35% more time to converge across pack types due to data duplication, although still falling within our time constraint. In Near-miss Under Sampling, we discarded majority samples (thereby throwing out data) to achieve class balance using Near-miss Algorithm [17]. This approach resulted in either PR-AUC parity or degradation of up to 7% across pack types while simultaneously reducing training time by as much as 40%. In the *Hybrid of Random Over Sampling and Random Under Sampling* approach, we evaluated on different hyper-parameters to randomly over sample the minority class and randomly under sample the majority class to achieve class balance.

Overall, the best performing parameters were skewed towards oversampling and resulted in 6%-10% improvement in model performance across pack types while increasing model training time by as much as 25%. In *Two-phase Learning with Random Under Sampling*, we used the model trained using under sampling as the pre-training phase and further fine-tuned the model on original imbalance data in the second phase. The first phase allows the minority group to influence the learning, and in the second phase the model still gets to see all data. This approach resulted in PR-AUC increase by 18%-24% across pack types while maintaining parity in training time compared to single modality baselines.

Algorithm Techniques In algorithm techniques, we modified the algorithm's loss functions to handle class imbalance, thereby allowing the minor samples to have more influence. We discarded cost-sensitive learning approaches as they did not satisfy our resource constraints. We tested two loss functions: Mean Squared False Error [19] and Focal Loss on the image modality alone [20]. Both approaches were easy to implement and caused no substantial change in training time. The mean squared false error loss function resulted in improving the model performance by as much as 14%. The focal loss functions have shown promising improvements on image datasets. The function reduces the importance of easy to classify samples thereby focusing on hard to classify samples. The focal loss improved model performance by 5% at best when implemented on the image modality component of our deep model (Table 1).

3. CONCLUSION

Our paper adds to sparse literature on how to use machine learning to choose a suitable pack type to ship a given product such that the products arrive undamaged, delight customers, and reduce packaging waste. Machine learning is crucial to make pack type decisions at scale given the hundreds of millions of products in a catalog. Approaches such as manual inspection of each product in the catalog to select a suitable pack type are non-scalable and impractical because both the catalog and pack types are dynamic; generic rule-based decisions (e.g., all product in "vinyl toys" category under $25 would go in a flexible mailer) fail to capture exceptions as they are not product-specific. Furthermore, our paper highlights what product-specific data we use is crucial to assess pack type suitability; especially including product images is important as text/tabular data lack information on how the product is currently packaged (e.g., LED bulb already packed in a box by the vendor) which can help reduce wasteful packaging. Synthesis studies on deep learning with class imbalance have found that the best technique to handle class imbalance varies by problem domain [7]. In fact, there exist few such studies that use data from either real-world applications or multimodal data [7]. In that regard, our learnings on imbalance handling techniques contribute to both packaging domain and broader literature on evaluating deep learning in the context of class imbalance and big data.

4. ACKNOWLEDGEMENTS

We thank Justine Mahler, Kim Houchens, and Nicola Preli for their insightful comments on the project. Opinions, findings, conclusions and recommendations expressed in this material are those of the authors and do not necessarily

reflect the views of Amazon.

References

[1] Venture Beat. *How Amazon is using machine learning to eliminate 915,000 tons of packaging*. https://venturebeat.com/2020/09/14/how-amazon-is-using-machine-learning-to-eliminate-915000-tons-of-packaging/. 2020.

[2] Fast Company. *Inside Amazon's quest to use less cardboard*. https://www.fastcompany.com/90564818/inside-amazons-quest-to-use-less-cardboard. 2020.

[3] Amazon. *Packaging*. https://www.aboutamazon.com/packaging. 2021.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. URL: https://doi.org/10.1038/nature14539.

[5] Anirudh Venkat Chari. "Prediction of Optimal Packaging Solution using Supervised Learning Methods". MA thesis. KTH, Mathematical Statistics, 2020.

[6] Dino Knoll et al. "An automated packaging planning approach using machine learning". In: *Procedia CIRP* 81 (2019), pp. 576–581. URL: https://doi.org/10.1016/j.procir.2019.03.158.

[7] Justin M. Johnson and Taghi M. Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (Mar. 2019), pp. 1–54. URL: https://doi.org/10.1186/s40537-019-0192-5.

[8] US Patent. *Predictive Packaging System. Application No: 16802876. Filed Date: 27 February 2020*. Country: United States of America. Feb. 2020.

[9] Jesse Davis and Mark Goadrich. "The Relationship between Precision-Recall and ROC Curves". In: *ICML '06: Proceedings of the 23rd international conference on Machine learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 233–240. ISBN: 1595933832. URL: https://doi.org/10.1145/1143844.1143874.

[10] Tom Zahavy et al. *Is a picture worth a thousand words? A Deep Multi-Modal Fusion Architecture for Product Classification in e-commerce*. https://arxiv.org/abs/1611.09534. Cornell University, 2016.

[11] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. https://arxiv.org/abs/1506.01497. Cornell University, 2015.

[12] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. URL: https://doi.org/10.1109/cvpr.2016.90.

[13] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, USA: IEEE, June 2009, pp. 248–255. URL: https://doi.org/10.1109/cvpr.2009.5206848.

[14] Piotr Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146. URL: https://www.aclweb.org/anthology/Q17-1010%22.

[15] R. Anand et al. "An improved algorithm for neural network classification of imbalanced training sets". In: *IEEE Transactions on Neural Networks* 4.6 (1993), pp. 962–969. URL: https://doi.org/10.1109/72.286891.

[16] Han H, Wang W-Y, and Mao B-H. "Borderline-smote: a new over-sampling method in imbalanced data sets learning". In: *Lecture Notes in Computer Science*. Ed. by Huang D-S, Zhang X-P, and Huang G-B. Adv Intell Comput. Berlin: Springer, 2005, pp. 878–887.

[17] I Mani and J. Zhang. "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction". In: *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*. Vol. 126. Washington, DC, United States: ACM, Aug. 2003, pp. 1–7.

[18] Hansang Lee, Minseok Park, and Junmo Kim. "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning". In: *2016 IEEE International Conference on Image Processing (ICIP)*. Phoenix, AZ, USA: IEEE, Sept. 2016, pp. 3713–3717. URL: 10.1109/ICIP.2016.7533053.

[19] S. Wang et al. "Training deep neural networks on imbalanced data sets". In: *International Joint Conference on Neural Networks, IJCNN 2016*. Vancouver, Canada: Institute of Electrical and Electronics Engineers (IEEE), Oct. 2016, pp. 4368–4374. ISBN: 9781509006212. URL: 10.1109/IJCNN.2016.7727770.

[20] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, Oct. 2017, pp. 2999–3007. URL: https://doi.org/10.1109/iccv.2017.324.