Deep Learning-Based Feature Extraction for Accurate Estimation of Product Dimensions from Images

Mohammed Moiz Pasha[1], **Mittapalli Pavan**[2], **Himanshu Parida**[3], **Mrs. K. Divyasri**[4]

[1] Department of CSE-AIML, Sreenidhi Institute of Science and Technology, India
[2] Department of CSE-AIML, Sreenidhi Institute of Science and Technology, India
[3] Department of CSE-AIML, Sreenidhi Institute of Science and Technology, India
[4] Department of CSE-AIML, Sreenidhi Institute of Science and Technology, India

---***---

Abstract - In the E-commerce sector, product dimensions play a crucial role both for inventory management and for enabling customers to filter products according to their requirements. Hence, accurately extracting dimensions from product data can help efficiently manage inventory space, save packaging costs and enhance customer experience. Existing methods use textual data like product title, description, and additional information for extracting dimension attributes. However, this textual information is usually ambiguous and disorganized. A few approaches also use images for dimension extraction, however these approaches only tackle length-based measurements, and do not provide an approach for handling other dimensions such as item weight, volume, and voltage. Hence, we propose a deep learning based approach to extract dimensions from product images, which uses Region Proposal Networks (RPNs) to create bounding boxes around products in images and Optical Character Recognition (OCR) pipelines with Regular Expressions (RegEx) to extract text bounding boxes. We then use relative positional information to identify the dimension in cases of length-based measurements. We showcase the results of our approach applied to a dataset of actual product images, and compare its performance with existing approaches.

Key Words: Dimension Extraction, OCR, RPN, Regular Expressions, Image Processing

1. INTRODUCTION

As e-commerce continues to evolve, product detail pages have become a vital bridge between sellers and consumers, conveying essential information through titles, descriptions, customer feedback, and images. To enhance product discovery, enable efficient comparison, and deliver a seamless shopping journey, online platforms are increasingly adopting scalable methods to derive structured product attributes—such as brand, color, and size—from the varied and often inconsistent data provided by sellers. The accuracy and completeness of these attributes directly influence search ranking algorithms, filter mechanisms, and recommendation systems. On the other hand, missing or inconsistent attributes can confuse shoppers, leading to abandoned purchases and higher return rates. Furthermore, incorrect dimensional attributes, such as weight, capacity, voltage, etc. can lead to supply chain issues and wastage of resources. Some e-commerce websites also allow users to sell products and directly upload images themselves. In such cases, moderation and maintaining accuracy of dimensions filled in by users becomes even more difficult, as it becomes difficult to monitor and keep track of products being sold.

Hence, an accurate and comprehensive system designed to automatically extract dimension-related attributes from product images can address multiple industry challenges. It can also enable new features to be built, such as image-based product searching, dosage determination from an image of medicines, etc. The proposed system aims to standardize dimension extraction processes, resulting in higher quality product catalogs and more efficient e-commerce operations, ultimately delivering a better experience for both businesses and consumers.

1.1 Related Works

There have been multiple works [5] [8] which aim to extract attributes from textual data using classical machine learning. Recent researches like OpenTag [1] applies Bidirectional-LSTM followed by conditional random field; similarly LaTeX-Numeric [2] extracts numeric attributes from textual data by solving a NER problem using BiLSTM-CNN-CRF model [3]. This, however, works only for text data, and not with images.

Zhou et al. [7] uses features from sections of the product images to improve recommendations. These works highlight the scope of the ample information that can be extracted from product images. However, the work is only used for recommendation systems, and not structured data extraction.

Ghosh et al. [4] extracts dimensional attributes from product images and applies a multi-box classification network based on transformer architecture, which predicts the length, width and height of a product provided. However, it deals with size attributes only, and not with other dimensions such as volume, wattage, etc.

2. PROBLEM DEFINITION

A product *p* in an e commerce website is described by an image *I* that contains dimensional information of the product as text within it. This dimensional information may represent the product’s height, width, length, weight, voltage, wattage, etc. The task is to extract the given dimensional attribute from the image based on its features, and convert it into a common format for interpretation.

For this implementation, we aim to extract the following dimensional attributes:

1. Width
2. Height
3. Length
4. Item weight
5. Item volume
6. Voltage
7. Wattage

3. METHODOLOGY

We propose a three-step processing pipeline to perform the dimension extraction process. This process comprises of an

OCR engine, which is used to extract the relevant textual information from the image along with its spatial information. The text boxes extracted from the image is passed to a *Postprocessor*, which uses RegEx to extract only dimensional information from the extracted text, and normalize it into a uniform representation. The image is then passed on to a *Region Proposal Network*, which identifies the position of the object in the image and returns the object's spatial context. The results of the above two are passed on to the *Dimension Resolution* stage, which combines all the spatial context and textual information, and uses statistical methods to extract the correct dimensional attribute from the image.

3.1 OCR engine

We deploy an OCR engine to extract the text and positional information from images. This OCR engine detect words and their bounding boxes in the images, and returns it along with the coordinates of the bounding boxes. For future computations, the coordinates of the centroid of each bounding box is considered as a representation of the location of the text with respect to the image.

3.2 RegEx Postprocessor

The text and bounding box coordinates extracted by the OCR engine is sent to the RegEx postprocessor. This postprocessor performs character-level substitution to convert the text and numeric dimension to a normalized format, for example, replacing characters such as " with the unit *"inch"*.
A RegEx lookup dictionary is created, which converts all representations of the unit to a chosen base format. This ensures that uniformity among units and dimensions is present, which is used further for comparison.

Base Unit	Lookup dictionary entry
"fluid ounce"	*"fluid ounce", "fl oz", "fluid ounces", "fl oz", "oz fl"*
"centimetre"	*"centimetre", "cm", "centimeter", "centimeters", "𝒄𝒎"*
"kilogram"	*"kilogram", "kg", "kilograms", "kgs", "kilogramme", "kilogrammes"*

Table -1: *Sample RegEx dictionary entries for specific units*

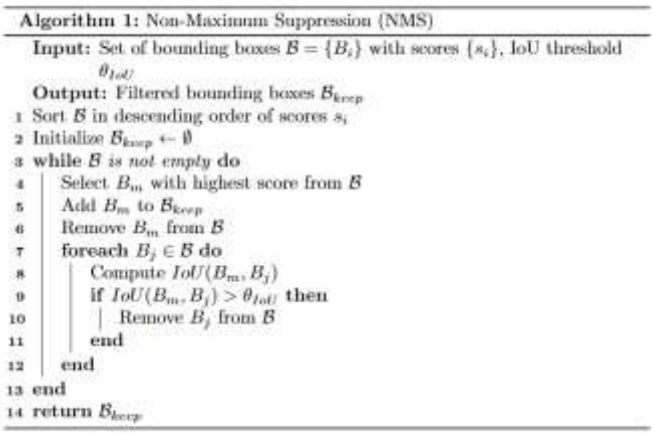
The postprocessor also ensures that the extracted numeric unit is properly formatted and converted to its decimal representation. For example, "*20 ½ in*" is converted to *"20.5 inch"*.

3.3 Region Proposal Network

A Region Proposal Network (RPN) is a fully convolutional network, which is used to predict potential regions of interest (bounding boxes) within an image. It identifies where to look for objects by generating a set of rectangular proposals and assigning them objectness scores [10]. RPNs are often used as a component of object detection algorithms like Faster R-CNN, enabling these algorithms to identify Regions of Interest (ROI) much more efficiently.
We use RPNs separately instead of traditional object detection algorithms since we do not require the additional context about the object itself and are only concerned with the spatial information about the position and size of the object in the image. Our RPN methodology hence enables us to fetch spatial coordinates of an object much faster, as the added overhead that comes with object classification is not used.
The RPN provides a set of bounding $\{B_i\}^N_{i=1}$ each with an associated objectness score $s_i \in [0,1]$ indicating the likelihood that the box contains an object. Each bounding box B_i is represented by its coordinates (x_i, y_i, w_i, h_i), or its corner format $(x_i^{(1)}, y_i^{(1)}, x_i^{(2)}, y_i^{(2)})$, where $(x_i^{(1)}, y_i^{(1)})$ denotes the top-left corner and $(x_i^{(2)}, y_i^{(2)})$ the bottom-right corner.\

3.3.1 Non-Maximum Suppression

After generating a set of candidate object bounding boxes from the Region Proposal Network (RPN), it is essential to reduce redundancy by selecting the most relevant and non-overlapping proposals. This is achieved through the application of Non-Maximum Suppression (NMS) [11], a technique that suppresses overlapping bounding boxes based on their objectness scores. NMS helps in retaining only the most confident bounding box among a group of overlapping proposals that correspond to the same object. NMS hence enables us to select the bounding box which best represents the spatial position of the object in the image.

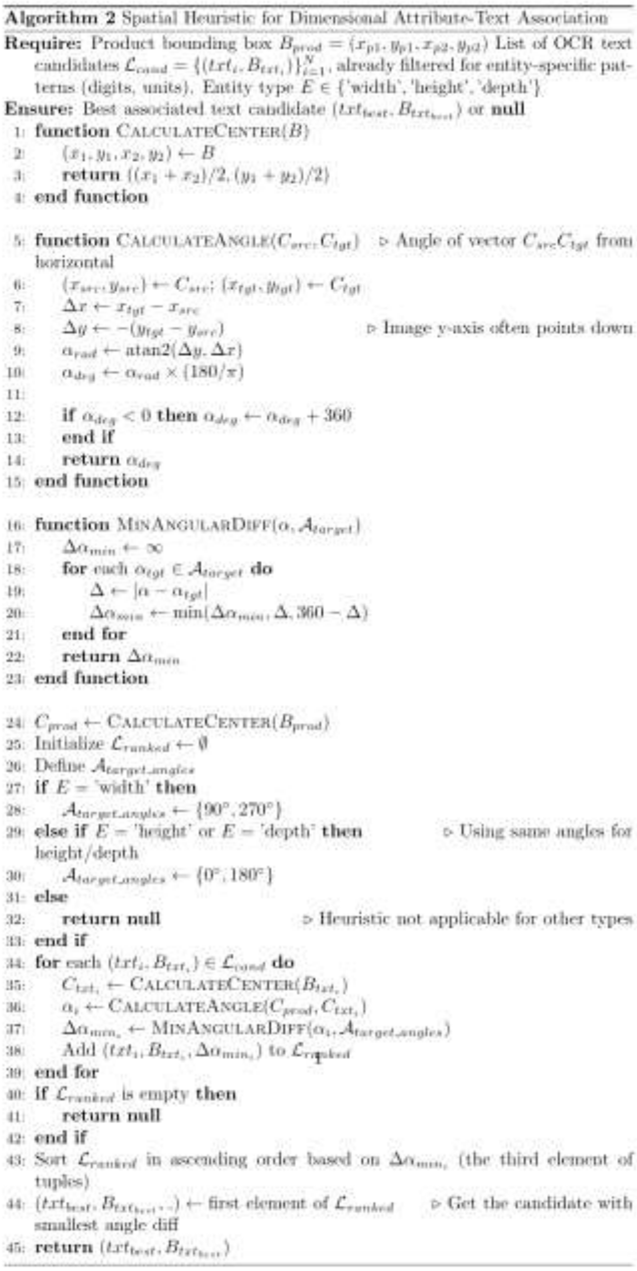


```
Algorithm 1: Non-Maximum Suppression (NMS)
Input: Set of bounding boxes $B = \{B_i\}$ with scores $\{s_i\}$, IoU threshold $\theta_{IoU}$
Output: Filtered bounding boxes $B_{keep}$
Sort $B$ in descending order of scores $s_i$
Initialize $B_{keep} \leftarrow \emptyset$
while $B$ is not empty do
    Select $B_m$ with highest score from $B$
    Add $B_m$ to $B_{keep}$
    Remove $B_m$ from $B$
    foreach $B_j \in B$ do
        Compute $IoU(B_m, B_j)$
        if $IoU(B_m, B_j) > \theta_{IoU}$ then
            Remove $B_j$ from $B$
        end
    end
end
return $B_{keep}$
```

Fig -1: Non-Maximum Suppression Algorithm

3.4 Dimension Resolution

After applying Non-Maximum Suppression (NMS), the retained bounding box is assumed to correspond to the most dominant object present in the image. This bounding box, along with its spatial coordinates and the output generated by the RegEx-based postprocessor, is leveraged to provide spatial context essential for the accurate extraction of dimensional attributes. For non-linear dimensions—such as weight, volume, voltage, wattage, and other scalar quantities—the resolution process involves identifying and aggregating text box values located in the immediate spatial vicinity of the object's bounding box. In such cases, the final predicted value is computed as the median of the values extracted from these nearby text boxes. This approach is motivated by the empirical observation that non-linear dimension values are typically positioned close to the centroid of the object's bounding box and are visually and semantically associated with the object as a whole. Hence, their proximity serves as a reliable indicator for correct value attribution. In contrast, linear dimensional attributes—such as length, width, height, depth, and similar spatial measurements— often correlate strongly with their respective orientations: for instance, length is generally annotated in alignment with the x-axis (horizontal), while height is typically associated with the y-axis (vertical). To exploit these positional regularities, the resolution of linear dimensions employs a custom-designed spatial heuristic algorithm.



```
Algorithm 2 Spatial Heuristic for Dimensional Attribute-Text Association
Require: Product bounding box $B_{prod} = (x_{p1}, y_{p1}, x_{p2}, y_{p2})$ List of OCR text
    candidates $\mathcal{L}_{cand} = \{(txt_i, B_{txt_i})\}_{i=1}^{N}$, already filtered for entity-specific pat-
    terns (digits, units). Entity type $E \in \{$'width', 'height', 'depth'$\}$
Ensure: Best associated text candidate $(txt_{best}, B_{txt_{best}})$ or null
function CalculateCenter($B$)
    $(x_1, y_1, x_2, y_2) \leftarrow B$
    return $((x_1 + x_2)/2, (y_1 + y_2)/2)$
end function

function CalculateAngle($C_{src}, C_{tgt}$)    ▷ Angle of vector $C_{src}C_{tgt}$ from
    horizontal
    $(x_{src}, y_{src}) \leftarrow C_{src}$; $(x_{tgt}, y_{tgt}) \leftarrow C_{tgt}$
    $\Delta x \leftarrow x_{tgt} - x_{src}$
    $\Delta y \leftarrow -(y_{tgt} - y_{src})$                    ▷ Image y-axis often points down
    $\alpha_{rad} \leftarrow \mathrm{atan2}(\Delta y, \Delta x)$
    $\alpha_{deg} \leftarrow \alpha_{rad} \times (180/\pi)$

    if $\alpha_{deg} < 0$ then $\alpha_{deg} \leftarrow \alpha_{deg} + 360$
    end if
    return $\alpha_{deg}$
end function

function MinAngularDiff($\alpha, \mathcal{A}_{target}$)
    $\Delta\alpha_{min} \leftarrow \infty$
    for each $\alpha_{tgt} \in \mathcal{A}_{target}$ do
        $\Delta \leftarrow |\alpha - \alpha_{tgt}|$
        $\Delta\alpha_{min} \leftarrow \min(\Delta\alpha_{min}, \Delta, 360 - \Delta)$
    end for
    return $\Delta\alpha_{min}$
end function

$C_{prod} \leftarrow$ CalculateCenter($B_{prod}$)
Initialize $\mathcal{L}_{ranked} \leftarrow \emptyset$
Define $\mathcal{A}_{target\_angles}$
if $E$ = 'width' then
    $\mathcal{A}_{target\_angles} \leftarrow \{90^\circ, 270^\circ\}$
else if $E$ = 'height' or $E$ = 'depth' then            ▷ Using same angles for
    height/depth
    $\mathcal{A}_{target\_angles} \leftarrow \{0^\circ, 180^\circ\}$
else
    return null                          ▷ Heuristic not applicable for other types
end if
for each $(txt_i, B_{txt_i}) \in \mathcal{L}_{cand}$ do
    $C_{txt_i} \leftarrow$ CalculateCenter($B_{txt_i}$)
    $\alpha_i \leftarrow$ CalculateAngle($C_{prod}, C_{txt_i}$)
    $\Delta\alpha_{min_i} \leftarrow$ MinAngularDiff($\alpha_i, \mathcal{A}_{target\_angles}$)
    Add $(txt_i, B_{txt_i}, \Delta\alpha_{min_i})$ to $\mathcal{L}_{ranked}$
end for
if $\mathcal{L}_{ranked}$ is empty then
    return null
end if
Sort $\mathcal{L}_{ranked}$ in ascending order based on $\Delta\alpha_{min_i}$ (the third element of
    tuples)
$(txt_{best}, B_{txt_{best}}, \_) \leftarrow$ first element of $\mathcal{L}_{ranked}$     ▷ Get the candidate with
    smallest angle diff
return $(txt_{best}, B_{txt_{best}})$
```

Fig -2: Spatial Heuristic algorithm for linear dimensions

The output of the algorithm represents the unit and value of the entity type. This is then transformed into a standardized format, which represents the resultant output of the entire pipeline.

4. EXPERIMENTS AND RESULTS

4.1. Dataset

For training and testing purposes, we use an official dataset of images provided by Amazon, containing ~260,000 images. The dataset consists of actual product images from Amazon as links, along with the *group_id*, *entity_name* which represents the dimension to be extracted, and the *entity_value* which is the actual value.

Fig -2: Sample images from dataset

Exploratory analysis of the dataset reveals that it contains a class imbalance towards the category of *item_weight*:

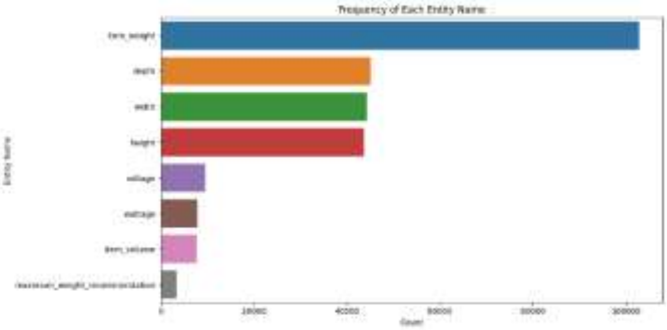


Fig -3: Frequency distribution of dataset by *entity_name*

This data is resampled, by undersampling the *item_weight* class in order to produce the final dataset. Further analysis reveals that the most common unit across all the records is “centimetre”, closely followed by “gram”.

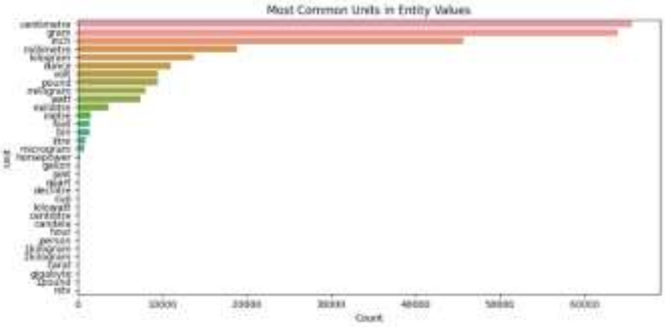


Fig -4: Frequency distribution of dataset by *units*

The dataset is partitioned to include a test set comprising approximately 131,000 images, which serves as the basis for the final evaluation of the pipeline's performance.
An important aspect to be noted is that actual entity values the source dataset by Amazon has been partially generated by their reinforcement learning algorithms, which may cause slightly inaccurate result representation.

4.2. Implementation Details

We implemented a PyTorch-based data loading framework in order to perform efficient batching and processing of the dataset, to enable parallel processing of multiple images at once. Each iteration of the DataLoader passes the input image to our pipeline. For the base OCR model in the OCR engine, we compared various open-source OCR models such as EasyOCR, PyTesseract, KerasOCR, etc. and found that PaddleOCR gave us the best results. For the RPN base network, the PyTorch's Feature Pyramid Network's (FPN) Region Proposal Network (RPN) – commonly a component of architectures like Mask R-CNN – was incorporated as a preliminary step. The rationale was to first localize the primary product or relevant regions of interest within the image. The RPN excels at generating candidate bounding boxes around potential objects, and the FPN architecture enhances this by effectively detecting objects across various scales, which is common in diverse product imagery.

4.3. Results

The application of our proposed pipeline to a large test dataset of approximately 131,000 images resulted in a strong F1-score of 0.58 for entity value extraction, highlighting the effectiveness of our multi-stage approach. This performance notably surpasses that of traditional computer vision methods typically used for similar tasks. Key to this success was the initial object localization using the Feature Pyramid Network's Region Proposal Network (RPN), which helped isolate the main product in varied image settings and enabled more accurate OCR results.

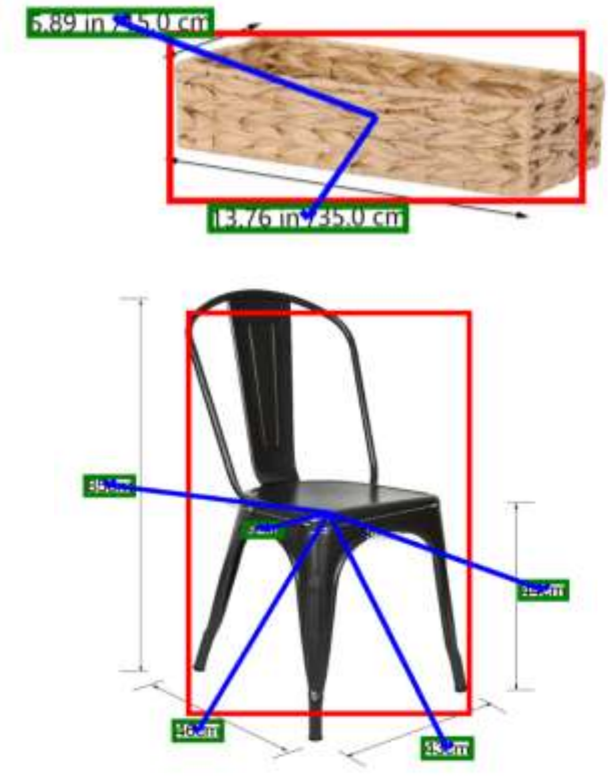


Fig-4 Visualization of detection results

Compared to emerging LLM-based approaches, our pipeline is over 50% faster and requires significantly less computational power. As shown in Figure 4, the system reliably detects key product boundaries and subtle dimension indicators, such as measurement lines, to extract relevant values. The seamless integration of object detection, focused OCR, and spatially-aware postprocessing underpins both the accuracy and efficiency of our method.

5. CONCLUSIONS AND FUTURE WORK

In this work, we proposed a comprehensive deep learning-based framework for the accurate extraction of product dimensional attributes from images, addressing a critical need in the e-commerce domain for structured and reliable product metadata. By integrating a Region Proposal Network (RPN) for spatial localization with a robust OCR pipeline and a RegEx-based normalization postprocessor, our method effectively extracts both linear and non-linear dimensions. The dimension resolution stage further enhances accuracy by incorporating spatial heuristics tailored to the orientation and proximity of dimensional text elements. Experimental evaluations on a large-scale dataset of real-world product images demonstrate the superiority of our approach in terms of F1-score and computational efficiency when compared to similar research works. The proposed pipeline not only generalizes well across various dimension types—including weight, volume, and voltage—but also presents a scalable and resource-efficient solution for industrial deployment. This work lays the foundation for future exploration into end-to-end vision-language systems for richer product attribute extraction.

Although our existing pipeline shows robust performance in extracting a broad spectrum of dimensional features, there is potential for improvement and extension. A potential direction would be to resolve overlapping or ambiguous regions of text at higher resolution with vision-language models that can more effectively comprehend contextual relationships between objects and text in scenarios that are complex. Embedding a lean, optimized transformer-based model may assist in resolving ambivalence when there are multiple dimension values, or ambiguous unit associations. The other direction to look at is semi-supervised or self-supervised learning methods in order to utilize better the huge amount of unlabeled product images from the internet. This would also enhance generalizability with minimal need for extensive human annotations. Finally, adding support for multilingual text recognition and finer-grained attribute types—like dosage for pharmaceuticals or energy rating for household appliances—would increase the versatility of the solution across world markets and industries.

ACKNOWLEDGEMENT

We would like to thank our guide Mrs. K. Divyasri, Associate Professor, Department of CSE - AIML and our coordinator Mr. G. Ravi, Assistant Professor, Department of CSE - AIML for their technical guidance, constant guidelines, encouragement, and support in carrying out this research work. We also profoundly thank Dr. K. Shirisha, Head of the Department of CSE - AIML, SNIST, who has been an excellent guide and a great source of inspiration to this work.

REFERENCES

[1] Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. Opentag: Open attribute value extraction from product profiles. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1049–1058, 2018.

[2] Kartik Mehta, Ioana Oprea, and Nikhil Rasiwasia. Latex-numeric: Language-agnostic text attribute extraction for e-commerce numeric attributes. arXiv preprint arXiv:2104.09576, 2021.

[3] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354, 2016.

[4] Ghosh, Pushpendu & Wang, Nancy & Yenigalla, Promod. (2023). D-Extract: Extracting Dimensional Attributes From Product Images. 3630-3638. 10.1109/WACV56688.2023.00363.

[5] Ghani, R., Probst, K., Liu, Y., Krema, M., & Fano, A. (2006). Text Mining for Product Attribute Extraction. ACM SIGKDD Explorations Newsletter, 8(1), 41–48.

[6] Lin, R., He, X., Feng, J., Zalmout, N., Liang, Y., Xiong, L., & Dong, X. L. (2021). PAM: Understanding Product Images in Cross Product Category Attribute Extraction. arXiv preprint arXiv:2106.04630.

[7] Wei Zhou, PY Mok, Yanghong Zhou, Yangping Zhou, Jialie Shen, Qiang Qu, and KP Chau. Fashion recommendations through cross-media information retrieval. Journal of Visual Communication and Image Representation, 61:112–120, 2019.

[8] Keiji Shinzato and Satoshi Sekine. Unsupervised extraction of attributes and their values from product description. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, pages 1339–1347, 2013.

[9] https://www.kaggle.com/datasets/suvroo/amazon-ml-challenge

[10] Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

[11] Neubeck, A., & Van Gool, L. (2006). *Efficient Non-Maximum Suppression*. 18th International Conference on Pattern Recognition (ICPR).