Ducho 2.0: Towards a More Up-to-Date Unified Framework for the Extraction of Multimodal Features in Recommendation

Matteo Attimonelli
Politecnico di Bari, Italy
matteo.attimonelli@poliba.it

Danilo Danese
Politecnico di Bari, Italy
danilo.danese@poliba.it

Daniele Malitesta*
Université Paris-Saclay,
CentraleSupélec, Inria, France
daniele.malitesta@centralesupelec.fr

Claudio Pomo
Politecnico di Bari, Italy
claudio.pomo@poliba.it

Giuseppe Gassi
Politecnico di Bari, Italy
g.gassi@studenti.poliba.it

Tommaso Di Noia
Politecnico di Bari, Italy
tommaso.dinoia@poliba.it

ABSTRACT

In this work, we introduce Ducho 2.0, the latest stable version of our framework. Differently from Ducho, Ducho 2.0 offers a more personalized user experience with the definition and import of custom extraction models fine-tuned on specific tasks and datasets. Moreover, the new version is capable of extracting and processing features through multimodal-by-design large models. Notably, all these new features are supported by optimized data loading and storing to the local memory. To showcase the capabilities of Ducho 2.0, we demonstrate a complete multimodal recommendation pipeline, from the extraction/processing to the final recommendation. The idea is to provide practitioners and experienced scholars with a ready-to-use tool that, put on top of any multimodal recommendation framework, may permit them to run extensive benchmarking analyses. All materials are accessible at: https://github.com/sisinflab/Ducho.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; **Personalization**.

KEYWORDS

Multimodal Recommendation, Deep Neural Networks

ACM Reference Format:
Matteo Attimonelli, Danilo Danese, Daniele Malitesta, Claudio Pomo, Giuseppe Gassi, and Tommaso Di Noia. 2024. Ducho 2.0: Towards a More Up-to-Date Unified Framework for the Extraction of Multimodal Features in Recommendation. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651440

1 INTRODUCTION AND MOTIVATIONS

Multimodal data sources (e.g., product images, descriptions, reviews, audio tracks) support recommendation systems in tasks such as fashion, micro-videos, and food recommendation. Indeed, the extraction of meaningful multimodal features from such data empowers the recommendation models by enriching their knowledge and understanding of users' preferences, eventually improving the quality of the proposed personalized suggestions. Nevertheless, to date, no standardized solutions for multimodal feature extraction/processing still exist in multimodal recommendation.

In our work [5], we introduced Ducho as a solution to unify the extraction and processing of multimodal features in recommendation systems. Ducho facilitates the creation of a comprehensive multimodal feature extraction pipeline, allowing users to specify data sources, backends, and deep learning models for each modality. The pipeline is easily configurable through a YAML file, simplifying the extraction and processing organized as sub-modules.

Even if the current functionalities enable to perform sufficiently extensive multimodal extraction pipelines to power the majority of existing multimodal recommender systems, we still recognize room for improvement in terms of the usability and optimization of the framework. Moreover, in the ever-evolving landscape of multimodal deep learning and recommendation, and (especially) with the recent outbreak of large models trained for several deep learning tasks, it becomes imperative to keep Ducho always up-to-date to implement and reflect such advances also in our framework.

Motivated by the outlined aspects, we present Ducho 2.0, the latest stable version succeeding Ducho. Our contributions focus on two main aspects: (i) improving the framework's **usability** and **customization** with **optimized** procedures, and (ii) enhancing multimodal feature extraction by incorporating recent advancements in **large multimodal models**.

Regarding the (i) contribution, Ducho 2.0 supports the adoption of custom extractor models with custom extraction layers, and the application of pre-processing operations (e.g., image normalization) whose parameters may be easily modified depending on the user's needs. Furthermore, towards efficiency, we introduce the popular PyTorch custom dataloader, that helps to optimize the overall loading/storing process. As for the (ii) contribution, we enrich the set of available backends-modalities configurations and implement the extraction/process of multimodal features through multimodal-by-design large models (such as CLIP [6]). In this respect, Ducho 2.0 is also capable of performing multimodal fusion operations.

The reminder of this paper elucidates each of the novelties introduced with Ducho 2.0. Additionally, we also propose a demonstration to showcase how Ducho 2.0 may be seamlessly used within a

[*] Work done while at Politecnico di Bari.

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0172-6/24/05.
https://doi.org/10.1145/3589335.3651440

complete multimodal recommendation pipeline, on top of the framework Elliot [1] tailored to address multimodal recommendation [4]. Indeed, we hope to provide practitioners and experienced scholars with an easy-to-use tool to run extensive benchmarking analyses with multimodal recommender systems, thus opening novel possible directions in the field. We release all the useful material for Ducho 2.0 at: https://github.com/sisinflab/Ducho.

2 NOVEL FEATURES

This section delves into Ducho 2.0, presenting new functionalities compared to Ducho [5]. The introduced features are categorized into two categories: (i) customization and optimization, and (ii) backends and large multimodal models. Figure 1 illustrates the architecture of Ducho 2.0, emphasizing the newly-added features.

2.1 Customization and optimization

• **Pipeline optimization.** Ducho 2.0 aims at enhancing computational efficiency. One significant addition is the implementation of multiprocessing facilitated by PyTorch-based dataloaders. Indeed, this new feature effectively leads to faster data loading and storing, which may represent a crucial bottleneck in the overall performance. Moreover, Ducho 2.0 is now capable of leveraging the computational speedup of MPS technology that alongside CUDA (already available in the previous version) makes our framework suitable across several existing platforms.

• **More flexibility and personalization.** Ducho 2.0 introduces a suite of novel features aimed at providing users with more customization options for the multimodal extraction pipeline. Noteworthy additions include the ability to specify desired types of pre-processing operations on images (i.e., z-score or minmax normalizations) that can be further customized through specific parameters. Moreover, users may now perform extractions using custom PyTorch and Transformers models for the **visual** and **textual** modalities, enhancing the flexibility and adaptability of the framework. In this respect, Ducho 2.0 also offers multiple image processors and tokenizers (which may be designed and pre-trained by the user) adding another useful level of customization.

2.2 Backends and large multimodal models

• **Additional backends settings.** Ducho 2.0 introduces advancements regarding backends. Notably, the Transformers backend is now available also for the **visual** modality together with the **textual** one (already present in the previous framework version). This novel feature comes as fundamental to enrich the extraction models' collection available in Ducho 2.0 through the extensive hub of pre-trained networks available on HuggingFace.

• **Multiple modalities and fusion.** Following the recent advances in large multimodal models, Ducho 2.0 now opens to the extraction and processing of multimodal features by directly using multimodal-by-design networks (e.g., CLIP [6]). Specifically, the framework integrates the **visual_textual** modality, building the necessary implementation bases for future extensions with other combined modalities. Then, users can require unified representations of the generated multimodal features by fusing their embeddings through various popular methods, such as concatenation, element-wise summation/multiplication, and averaging.

3 THE NEW CONFIGURATION FILE

To provide a more technical description of Ducho 2.0, we show a toy YAML configuration file (Configuration 1) which spans all the core functionalities introduced in Ducho 2.0. Note that, to support the seamless transaction from Ducho to Ducho 2.0, the configuration file retains the same modular structure as before.

First, in the **visual** extraction, users can set the `preprocessing` procedure to perform (i.e., either z-score or minmax, optionally with custom `mean` and `std` values) in the case of PyTorch backend. Under the same backend setting, the configuration may also include the loading of a custom extraction model; note that the user will need to indicate the exact path to the pre-trained weights (i.e., the `.pt` file) and include or import the definition of its architecture as a `torch.nn.Model` in the main Python script. Similarly, when leveraging the Transformers backend, it is now possible to specify a custom `model_name`, as the framework will first search on the HuggingFace hub and (if not available) at a local path, allowing to utilize custom pre-trained models. Noteworthy, the `image_processor` and the `output_layers` may be also customized.

Then, the **textual** extraction permits to load pre-trained tokenizers from HuggingFace or rely on own specified tokenizer using the `tokenizer_name` parameter. In addition to this, we have fixed some code bugs regarding the names of the `.tsv` columns standing for the items' IDs and their description; thus, it is now possible to explicitly indicate custom column identifiers.

Finally, in the novel scenarios involving multiple modalities (e.g., **visual_textual**), it is worth underlining that fusion techniques are also available to combine the extracted **visual** and **textual** modalities into a single representation. Currently, Ducho 2.0 supports concatenation (i.e., `concat`), as well as element-wise operations such as summation (i.e., `sum`), multiplication (i.e., `mul`), and average (i.e., `mean`). Note that, in such cases, the framework will check for dimensionalities mismatch, as for element-wise operations the two fused vectors are supposed to have the same embedding size. Moreover, output folders for the fused multimodal features will be named accordingly (e.g., `vis_embeddings_text_embeddings_concat`).

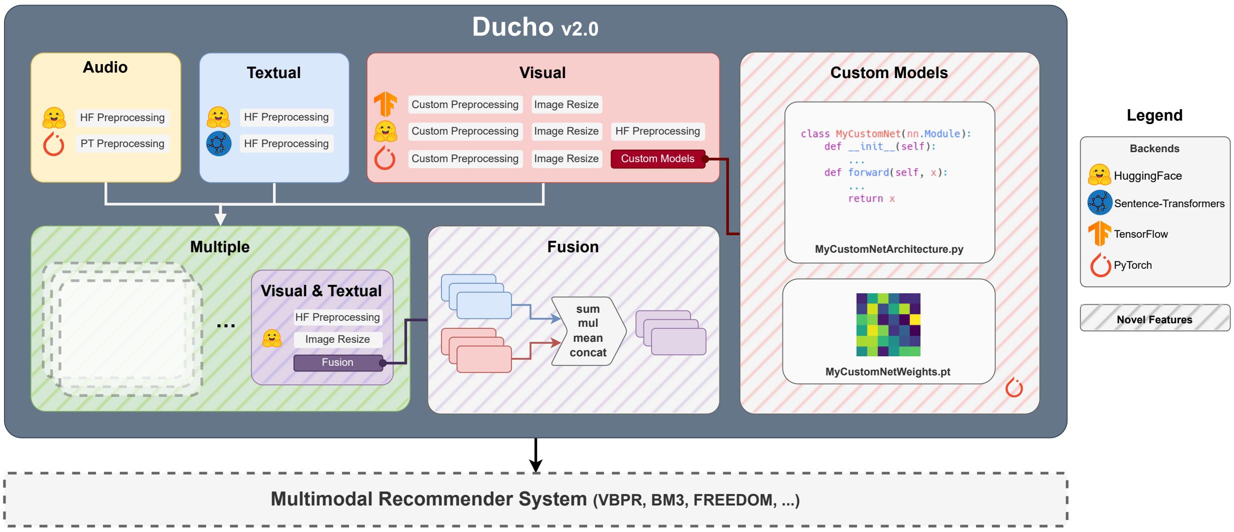
4 MULTIMODAL BENCHMARKS

This section highlights Ducho 2.0 into a multimodal recommendation pipeline, demonstrating its usability as a feature extractor for benchmarking any multimodal recommender system. We show two new functionalities: (i) feature extraction through multimodal-by-design models, and (ii) custom models. We outline the construction of the multimodal dataset, describe the feature extraction, present the multimodal recommendation approaches, explain how to run the demonstration and discuss the results.

4.1 Dataset preparation

We use **Amazon Baby** from the Amazon recommendation dataset[1]. Recorded items are enriched with metadata, with their unique identifiers, textual descriptions, and image URLs for product photos. In the preprocessing phase, we retain products with valid image URLs and non-empty textual descriptions, representing the **Visual** and **Textual** modalities in our recommendation scenario. At the

[1] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html.



```
class MyCustomNet(nn.Module):
    def __init__(self):
        ...
    def forward(self, x):
        ...
        return x
```

Figure 1: An overview of Ducho 2.0, where newly-introduced functionalities have hatch background.

```
dataset_path: ./my/dataset/path
gpu list: 0
visual:
 items:
  input_path:  images
  output_path: visual_embeddings
  model: [
    { model_name: ResNet18, output_layers: avgpool,
      reshape: [224, 224], backend: torch, preprocessing : zscore,
      mean : [0.485, 0.456, 0.406], std : [0.229, 0.224, 0.225] },
    { model_name: ./MyCustomNetWeights.pt , backend: torch,
      output_layers: pooler_output, preprocessing : minmax },
    { model_name: ./MyCustomHFModel , backend: transformers ,
      output_layers: [MyCustomOutputLayer , avgpool],
      image_processor : ./MyCustomImageProcessor } ]
textual:
 items:
  input_path:  descriptions.tsv
  output_path: textual_embeddings
  item_column: asin
  text_column: description
  model: [
   { model_name: ./MyCustomHFModel , clear_text: False,
     output_layers: MyCustomOutputLayer , backend: transformers,
     tokenizer_name : ./MyCustomTokenizer } ]
visual_textual :
 items:
  input_path:  { visual: images, textual: meta.tsv }
  output_path: { visual: vis_embeddings, textual: text_embeddings }
  item_column: asin
  text_column: description
  model: [
   { model_name: openai/clip-vit-base-patch16, fusion : concat,
     output_layers: 1, backend: transformers  } ]
```

Listing 1: A toy example with the YAML configuration. New features for Ducho 2.0 are highlighted in green.

time of this submission, we have 6,386 valid items, 19,440 users, and 138,821 recorded user-item interactions.

4.2 Multimodal feature extraction

Our benchmarking analysis covers three multimodal feature extraction settings. Firstly, in a common multimodal recommendation scenario, we extract **Visual** and **Textual** features using pre-trained ResNet50 and SentenceBert, resulting in 2048- and 768-dimensional embeddings, respectively. Secondly, we introduce a novel feature in Ducho 2.0 by employing a pre-trained multimodal model (CLIP) with ViT-B/16, producing 512-dimensional embeddings for both modalities. Thirdly, we showcase another Ducho 2.0 functionality: the extraction of multimodal features through a pre-trained custom model, MMFashion[2] [3]. For the **Visual** modality, it uses a fine-tuned ResNet50 backbone for fashion attribute prediction, yielding 2048-dimensional embeddings. For the **Textual** modality, we adopt SentenceBert, resulting in 768-dimensional embeddings.

4.3 Multimodal recommendation

We use three multimodal recommendation approaches: VBPR [2, 8], BM3 [10], and FREEDOM [9]. We train and test the three recommendation models by adopting their re-implementations within Elliot [1]; you may consider this public repository[3] for a reference of the models' codes and settings. Technically, we start from the whole user-item interaction data and perform a 80%/20% random hold-out split for the training and test sets, retaining the 10% of the training as validation. Then, we perform a grid search hyper-parameter exploration for each model by following the same experimental setting proposed in [4], where the Recall@20 is used as a validation metric. Note that each of the three multimodal recommendation approaches is trained and tested all over again for each multimodal feature extraction setting as reported above.

[2] https://drive.google.com/open?id=1LmC4aKiOY3qmm9qo6RNDU5v_o-xDCAdT.
[3] https://github.com/sisinflab/Formal-MultiMod-Rec.

Table 1: Recommendation results with varying multimodal feature extractors on top-20 recommendation lists.

Extractors	Models	Recall	Precision	nDCG	HR
Visual: ResNet50 **Textual**: SentenceBert	VBPR	0.0613	0.0055	0.0309	0.1031
	BM3	0.0850	0.0076	0.0426	0.1420
	FREEDOM	0.0935	0.0083	0.0465	0.1537
Visual & **Textual**: CLIP	VBPR	0.0630	0.0057	0.0313	0.1068
	BM3	0.0851	0.0076	0.0428	0.1425
	FREEDOM	0.0704	0.0064	0.0351	0.1187
Visual: MMFashion **Textual**: SentenceBert	VBPR	0.0619	0.0055	0.0309	0.1027
	BM3	0.0847	0.0076	0.0420	0.1423
	FREEDOM	0.0932	0.0083	0.0465	0.1542

4.4 Running the multimodal benchmarks

We provide three ways to run the multimodal benchmarks: (i) locally, through the GitHub repository; (ii) on Google Colab, through a well-documented Jupyter Notebook[4]; (iii) by instantating and running a Docker container through Ducho 2.0's Docker image[5] and a new Docker image equipped with Elliot for multimodal recommendation[6]. We especially suggest following either (ii) or (iii) since they require few installation steps. Comprehensive documentation is accessible online[7].

4.5 Results

Table 1 shows recommendation results in the three multimodal settings reported above on the retrieved Amazon Baby dataset. Note that all considered metrics account for top-20 recommendation lists.

Overall, we notice a general trend across all multimodal feature settings, namely, more recent recommendation approaches perform better than previous solutions in the literature (FREEDOM and BM3 always outperform VBPR on all the recommendation measures). Then, on a finer-grained evaluation regarding each multimodal feature setting, we observe that the adoption of pre-trained multimodal extractors (i.e., CLIP) or vision deep networks fine-tuned on the fashion domain (i.e., MMFashion) may provide improved recommendation performance in some cases. Specifically, compared to the standard setting including ResNet50 and SentenceBert as extractors, VBPR and BM3 generally reach higher accuracy metrics when leveraging CLIP as multimodal feature extractor (for VBPR, this has been shown in [7]) or MMFashion for the **Visual** modality.

However, the observed trends regarding multimodal feature extractors may not be easily generalized; for instance, there are settings when we recognize a drop in performance with respect to the standard multimodal setting. For instance, consider the consistent performance drop of FREEDOM when using CLIP. Such results call for more careful analyses regarding the possible impact of recent multimodal large models used as feature extractors in multimodal recommendation [7]. Indeed, this novel evaluation is out of the scope of this work, but we deem Ducho 2.0 may be efficiently and effectively utilized to conduct extensive benchmarking analyses.

5 CONCLUSION AND FUTURE WORK

This paper introduces Ducho 2.0, the new version of our framework Ducho for the unified extraction and processing of multimodal features for multimodal recommendation. Building upon its predecessor, Ducho 2.0 includes: (i) framework customization and optimization and (ii) the introduction of novel backend-modality settings and large multimodal models. On the one hand, Ducho 2.0 facilitates the integration of personalized extractor models featuring custom extraction layers, along with the application of preprocessing tasks like custom image normalization. To enhance the data loading and storing, we exploit the widely-used PyTorch custom dataloader. On the other hand, we expand the suite of available backend-modality configurations and implement multimodal-by-design extraction models, allowing modality fusion. To test the new functionalities, we use Ducho 2.0 on top of a multimodal recommendation framework, opening to extensive benchmarking analyses. We plan to further enhance the framework's optimization through multi-GPU extractions and integrate other multimodal-by-design models. Moreover, we intend to use Ducho 2.0 to bring our contribution to multimodal recommendation.

ACKNOWLEDGMENTS

This work has been carried out while *Matteo Attimonelli* was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with *Politecnico Di Bari.* This work was partially supported by the following projects: CT_FINCONS_III, OVS Fashion Retail Reloaded, LUTECH DIGITALE 4.0, VAI2C, IDENTITA, REACH-XY. We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support.

REFERENCES

[1] Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. 2021. Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. In *SIGIR*. ACM, 2405–2414.

[2] Ruining He and Julian J. McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI*. AAAI Press, 144–150.

[3] Xin Liu, Jiancheng Li, Jiaqi Wang, and Ziwei Liu. 2021. MMFashion: An Open-Source Toolbox for Visual Fashion Analysis. In *ACM Multimedia*. ACM, 3755–3758.

[4] Daniele Malitesta, Giandomenico Cornacchia, Claudio Pomo, Felice Antonio Merra, Tommaso Di Noia, and Eugenio Di Sciascio. 2023. Formalizing Multimedia Recommendation through Multimodal Deep Learning. *CoRR* abs/2309.05273 (2023).

[5] Daniele Malitesta, Giuseppe Gassi, Claudio Pomo, and Tommaso Di Noia. 2023. Ducho: A Unified Framework for the Extraction of Multimodal Features in Recommendation. In *ACM Multimedia*. ACM, 9668–9671.

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 8748–8763.

[7] Zixuan Yi, Zijun Long, Iadh Ounis, Craig Macdonald, and Richard McCreadie. 2023. Large Multi-modal Encoders for Recommendation. *CoRR* abs/2310.20343 (2023).

[8] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining Latent Structures for Multimedia Recommendation. In *ACM Multimedia*. ACM, 3872–3880.

[9] Xin Zhou and Zhiqi Shen. 2023. A Tale of Two Graphs: Freezing and Denoising Graph Structures for Multimodal Recommendation. In *ACM Multimedia*. ACM, 935–943.

[10] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2023. Bootstrap Latent Representations for Multimodal Recommendation. In *WWW*. ACM, 845–854.

[4] http://tinyurl.com/mpvv39rp.

[5] https://hub.docker.com/repository/docker/sisinflabpoliba/ducho/.

[6] https://hub.docker.com/repository/docker/sisinflabpoliba/mm-recsys/.

[7] https://github.com/sisinflab/Ducho/tree/main/demos/demo_recsys/README.md.