Deep Multimodal Learning for E-commerce Product Representations

Wing-kit Chan
Chun-kiu Kwok

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong
19 April 2023

Contents

List of Figures

List of Tables

List of Algorithms

Deep Multimodal Learning for E-commerce Product Representations

AIST4999 Final-Year Project II, 2022/23 Term 2 Thesis Report

Wing-Kit, Chan
AIST, 1155142198@link.cuhk.edu.hk

Chun-Kiu Kwok
AIST, 1155141911@link.cuhk.edu.hk

Abstract

E-commerce has become an attractive market strategy for expanding and automating businesses. To allow a quick and satisfactory consumer experience, e-commerce platforms must devise effective data representations to manage an ever-expanding number of products while maintaining high-quality retail services. Recent studies from technological giants have also shown great interest in using unimodal and multimodal neural networks in e-commerce. Although unimodal approaches using images or texts were already investigated, the use of multimodal approaches is still in its infancy, with papers published in the previous one or two years.

In this report, we adopt the Visual and Language Transformer (ViLT) model for generating product embeddings from 3 different public product datasets: Fashion Products Image Dataset, Farfetch Fashion Outfit Challenge, and the Shopee Price Match Guarantee. We also developed three visual models, three textual models and a late fusion multimodal model for comparison purposes. We show that our ViLT model has matched or outperformed baseline visual and textual models on the datasets, proving our postulate that the multimodal models produce more precise embeddings by utilizing information from several modalities.

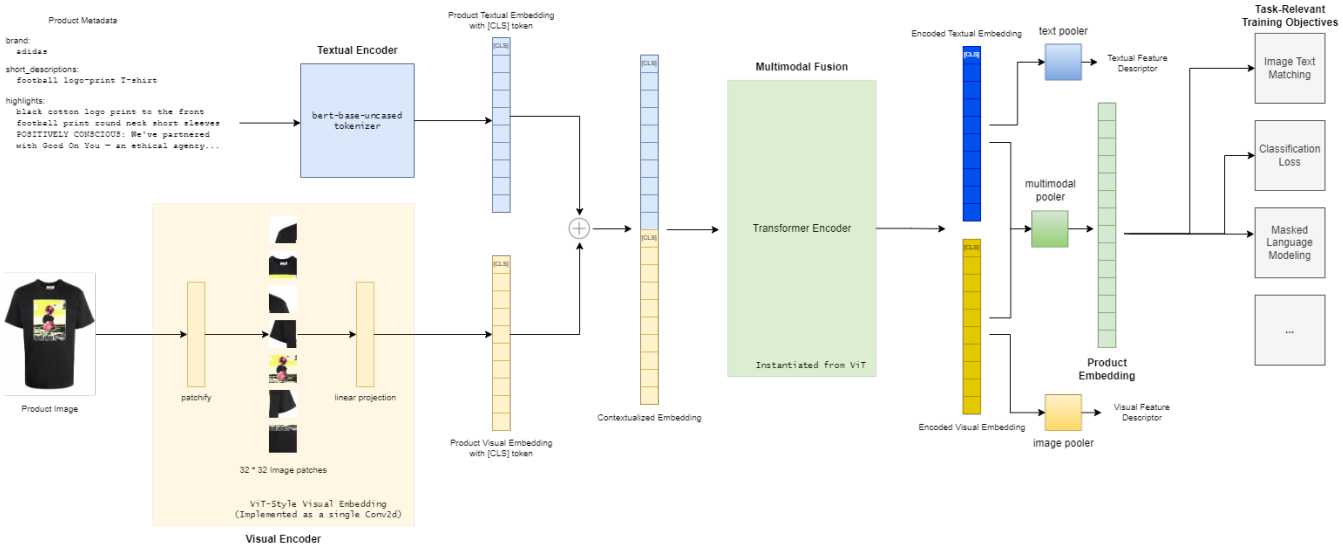


Figure 1: Overview of our multimodal modelling approach. Product images and their respective metadata are forwarded to the visual and textual encoder separately to generate product visual and textual embeddings, which are then passed to the transformer encoder for multimodal fusion. The resulting product embeddings will be applied to different task-relevant learning objectives. Textual and visual feature descriptors may also be extracted from the encoded embeddings and adopted according to the learning objective.

1 Introduction

As e-commerce becomes a trending marketing strategy, many retail enterprises aim at expanding their business using the Internet to reach more potential customers through advertisements and online stores. Moreover, as intelligence systems are getting more attention, automation is also a direction of improvement. With the increasing popularity of cryptocurrencies and metaverse, most purchases will likely occur online soon. Under this trend, e-commerce platforms must provide innovative online shopping services while managing the fast-growing volume of products. Effective product data representations will thus be the key to optimizing online service quality and storage management, enabling a swifter and more satisfactory shopping experience.

Embeddings refer to the low-dimensional projections of high-dimensional feature vectors at which the main input characteristics are still preserved [3]. They are commonly used in natural language processing tasks in the form of word embeddings, representing the sparse word vectors in a more compact space that favours the training process. Over the past years, embeddings have been generalized to represent different kinds of sparse vectors, such as Item2vec, MetaPath2vec, and Prod2vec. Product embedding is a specific application of embeddings, in which product data are encoded by dimension-scalable vectors based on their information like item description.

Modality is generally described as "the way in which something happens or is experienced," and can be understood as the data types of features in the machine learning field [5]. Unimodal learning approaches only use a single data type in model training, such as image, text, or audio. Due to their simplicity, models developed using this approach are easy to build and train but are also more susceptible to noisy data. On the other hand, since multimodal learning needs to combine two or more data types, models built using this approach are much harder to interpret and require specific model designs to develop but are more robust against noise due to the increased amount of information.

In this project, multimodal learning approaches will be applied to convert product thumbnails and descriptions into compressed representative vectors known as product embeddings. Product embeddings encapsulate the distinct features of each product in compact forms, efficiently grouping similar products while distinguishing dissimilar products. As a more precise and convenient data representation method, product embeddings can be utilized in several e-commerce downstream tasks such as categorization, price prediction, and product matching. The project highlights are to analyze how well product embeddings can be utilized in the e-commerce business and to compare the downstream task performance between multimodal product embedding approaches and typical unimodal methods. Kwok is responsible for all unimodal model implementation and training while Chan is responsible for Vision-and-Language Transformer training and dataset preparations. A detailed labour distribution table can be found in Appendix C.

2 Related Work

Researchers have shown increasing interest in product embeddings in the last few decades. Various image-based, text-based, and multimodal approaches have been proposed for generating efficient representations of e-commerce product catalogues.

2.1 Image-based and text-based approaches

Image-based approaches only consider the products' images when generating the representative embedding. Put simply, it is the same as classifying images into their corresponding classes. For instance, convolutional networks are deployed on social media images to predict hashtags and deploy such a model for fine-tuning in other downstream tasks such as image classification and object identification [22]. Originally conducting a study on transfer learning, the researchers utilized ResNeXt models with 101 layers of different configurations (see Section 5 for a detailed explanation) and then pre-trained their models on Instagram images that were labelled with various hashtags and attempted to use these models on ImageNet classification tasks and the Common Objects in Context (COCO) Captions dataset, with at least 85% top-1 accuracy on ImageNet. A year later, a large-scale commerce search engine named MSURU is developed which is deployed on Facebook Marketplace to process product images for searching and recommendation [32]. ResNeXt-101 is adopted and fine-tuned the model on the search query data on Facebook Marketplace. This MSURU model is reported to have outperformed the ResNeXt-101 by 16% in the e-commerce domain and made enormous improvements in search success rate and active interaction. In these examples, the models are often used to classify images and aid image product searching, which proves the point that there is a great demand for such object identification models in the field of e-commerce.

Similarly, text-based approaches only consider the products' descriptions and metadata. These descriptions and metadata may include titles, a list of functionalities or features. For example, Meta-Prod2Vec, an improved version of Prod2Vec and ultimately a derivation of Word2Vec, is proposed for generating low-dimensional embeddings of the products [33]. Meta-Prod2Vec utilizes past user interactions, reviews, and metadata to produce product embeddings and provide useful recommendations to boost sales. Their paper showed that Meta-Prod2Vec achieved a hit rate 6 times higher than only using simple metadata like popularity or co-occurrence with other products. Word2Vec embeddings are also proposed to help retailers match their products with their competitors' products to provide customers with another choice and gain the most profit [21]. This method uses the fastText pre-trained word embedding, an embedding based on the skip-gram model, together with other measures of similarity to construct a product matching model using an extreme gradient boosting ensemble (XGBoost), a more efficient algorithm based on decision trees, on two product catalogues taken from European Union-based consumer electronics retailers. The most important takeaway is that the trained XGBoost model can match 81% of the 38 obtained matches

correctly, even though the two product catalogues chosen for testing were extremely imbalanced and did not have a consistent category tree. These attempts also inspired our investigation of using more complex natural language processing networks in the field of e-commerce to help customers find their favourite products and increase the company's competitive edge.

2.2 Multimodal Fusion

Multimodal fusion in machine learning refers to the techniques used to aggregate information from different modalities of data, such as text, audio, and image. Based on the stage at which the aggregation takes place, multimodal fusion strategies can be categorized into early fusion, joint fusion, and late fusion [16, 26]. Early fusion approaches combine raw or preprocessed modalities into a single vector before forwarding to the main model. The combination of modality features can be performed in several ways. For instance, Poria et al. [24] concatenate textual, visual, and audio feature vectors from short video clips to predict the sentiment polarity of the speaker, while Douwe et al. [18] apply component-wise maximum and gated units to fuse image and textual inputs from multiple datasets. Similar to early fusion, joint fusion techniques combine the input modality feature vectors and also the intermediate features from the main model. On the other hand, late fusion strategies aggregate the predictions made by multiple single modality models for generating the final decision [16, 26]. This form of decision-level fusions can be achieved with parameter-free methods such as voting and weighted sum [23, 25, 31, 37], or more complex structures such as extra classifier networks [27, 41]. In order to obtain representative vectors from cross-modality data, the choice of multimodal fusion strategy must adapt to the dataset characteristics and the task nature for extracting different pieces of task-relevant information from different feature modalities.

Applications of multimodal fusion can be found in vast varieties of fields, such as image captioning [11, 19, 38], video description [15], semantic segmentation and image-to-image translation [35]. For image captioning, [38] trains an image encoder for image feature extractions and a shared Transformer model for both text encoding and multimodal fusion by applying captioning loss over the multimodal text decoder, while [11, 19] merges the encoded image and text features via the co-attention blocks of an individual Transformer encoder with several training objectives. This variant of the Transformer model for processing image and text features is termed the Vision-and-Language Transformer (ViLT). Similar to the above works, [15] utilizes the attention mechanism to incorporate audio, image, and motion features for generating video summaries. On the other hand, [35] proposes a parameter-free multimodal fusion framework for segmentation by exchanging channel features of different modalities and by sharing a learnable convolutional filter across modalities. Compared with models that train on a single modality, deep multimodal fusion models learn the relationship between different modality inputs and tend to produce

more accurate predictions [11, 15, 19, 35, 38].

Most of the research on multimodal product fusion emphasizes the aggregation of middle–level features, i.e., joint fusion. [12] concatenate the processed product text, image, and category features, and forward the resulting vectors to fully connected layers for price prediction. [28] include two separate subnetworks to retrieve each fashion product's image and text descriptors, and then apply contrastive loss over the descriptors to train for joint product embeddings. [29] presents a joint embedding model consisting of a product image encoder and a product metadata encoder. The mini-batch match retrieval objective function is also suggested to align the embeddings produced by the encoders. [39] proposes a series of cross-modality and cross-product-pair pre-training tasks, termed Omni Retrieval, for multimodal representation learning with the Transformer models on 110 million product-related image-text pairs. Inspired by the success of Vision-and-Language Transformer (ViLT) in image captioning [11, 19], we will explore the application of Transformer-series models in product embedding generation and discuss their capabilities of merging cross-modal information.

3 Method

We compare the multimodal learning approach with other unimodal model baselines on the task of product embedding generation. For the visual models, we utilize the residual network (ResNet), the ResNeXt model, and the Vision Transformer (ViT) for visual inputs. At the same time, we use the Bidirectional Encoder Representations from Transformers (BERT) model, the DistilBERT model and the DistilRoBERTa model to process textual inputs. For the multimodal learning approach, we adopt the Vision-and-Language Transformer ViLT to integrate both textual and visual product information into the embedding generation [11]. We further reference the multimodal End-to-end Transformer framework to diagnose and train the ViLT model in an end-to-end manner [19]. A late fusion model is also developed to compare with the ViLT model. To optimize the model performance, we experiment with several objectives and loss functions during the training process, namely triplet loss and Omni Retrieval.

3.1 Image-based Models

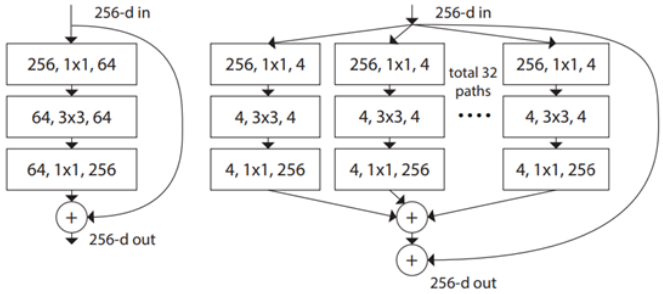


Figure 2: Left: A bottleneck building block for ResNet-50. Right: A block of ResNeXt with cardinality = 32. Source: [36].

3.1.1 ResNet

The ResNet model is a highly modularized model in which the layers are encapsulated into residual blocks [13]. These blocks are composed of convolutional layers, usually with a filter size of 3 * 3. However, different from the plain convolutional networks, the resultant tensor is added to the original input. This is termed the "shortcut" because it allows information to bypass any unnecessary convolutional layers through the summation. With such implementation, the authors proved that ResNet achieves lower error rates than plain models that only stack convolutional layers. In addition to the residual block, they modified the residual block with the bottleneck design to attempt to build deeper residual networks. These bottleneck modules take in a 256-channel tensor and pass it through three convolutional layers with filters of sizes 1 * 1, 3 * 3 and 1 * 1, each followed by a batch normalization layer, and

return a tensor of the same size as the input. To reduce the number of computations, the intermediate output is diminished to a size of 64 * 64, and that is the reason why the module is named "bottleneck". With these two design changes, they then attempted to train a 1202-layer ResNet model with "no optimization difficulty" and the model achieved over 99.9% training accuracy with a testing error of a little less than 8%. The birth of ResNet became a breakthrough due to its ability to train extremely deep models.

3.1.2 ResNeXt

A more advanced version called the ResNeXt further extends the modularity of the model by combining ResNet and the mechanism of Inception (split-transform-merge), which helps reduce the number of parameters and hyperparameters. Inspired by simple neurons, the term "aggregated transformations" is defined, which mimics the effect of inner products, but instead of multiplying a vector by a weight vector, the vector undergoes a transformation [36]. This motivates the authors to introduce a new dimension — the number of branches used during the split — inside the bottleneck module. This number of branches is called "cardinality", which is the number of "set of transformations" to be aggregated. Although the transformations can be arbitrary, they opted for setting all transformations to the same configuration for simplicity and the sake of a fair experiment. Similar to ResNet, the image is passed into the bottleneck block composed of filters of sizes 1 * 1, 3 * 3 and 1 * 1 respectively; however, the width of the bottleneck is reduced from 64 to 4 to cater for the newly introduced cardinality. The result of each branch is first aggregated and then summed with the original tensor, which resembles the mechanism of ResNet. With the introduction of cardinality, the authors proved that ResNeXt achieves better accuracy without drastically complicating the model, thus simpler but shallower models can be developed.

3.1.3 Vision Transformer

However, most computer vision models rely on the structure of CNNs or make similar assumptions or "inductive biases" such as locality and two-dimensional neighbourhood structure, which restricts CNNs' ability to capture global features. It is suggested that this reliance can be scrapped and, to mitigate this defect, a non-convolutional model termed the Vision Transformer is developed [10]. The Vision Transformer mimics the Transformer, a renowned model in the field of natural language processing. The authors divide the input image into patches of equal sizes to achieve this effect. Similar to Bidirectional Encoder Representations from Transformers (BERT), an extra learnable class embedding is also prepended to the embedded patches to facilitate classification. They then linearly embed each patch and add one-dimensional position embeddings to retain positional information as a sequence. This resulting sequence of vectors is then fed into a Transformer encoder composed of blocks containing two parts. The

first part is a layer normalization layer followed by a multi-head attention layer while the second part is a layer normalization layer followed by two fully connected layers with a Gaussian Error Linear Unit (GELU) activation. Finally, the output of the encoder is fed into a classification head for class prediction, which can be a linear layer or multilayer perceptron. This implementation has limited the assumption of locality and two-dimensional neighbourhood structure and instead enables the model to generalize better.

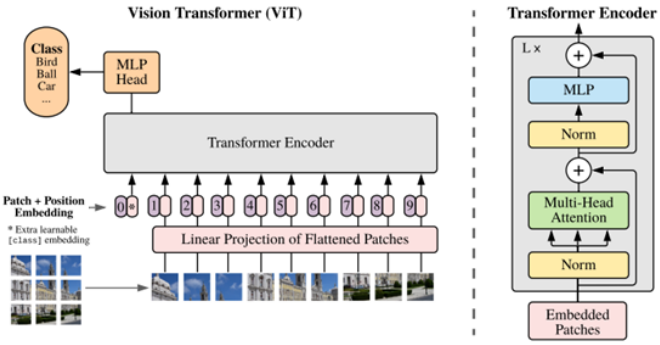


Figure 3: Model overview of Vision Transformer. Source: [10]

To sum up, although these models have different scopes of inductive biases, they all extract the information from the visual input to create a product embedding. These models are chosen because they have a modest number of parameters without being too simple or too complex for our task (see Tables 1 and 2 for the numbers of parameters), which also helps us to control our experiments better without having to manage an enormous amount of hyperparameters.

3.2 Text-based Models

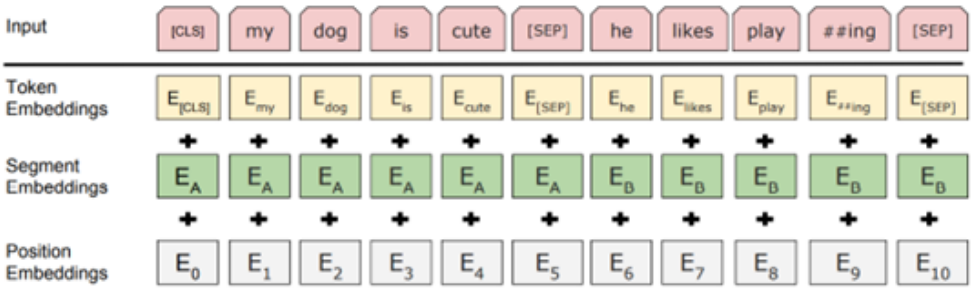


Figure 4: Model overview of BERT. Source: [9]

3.2.1 BERT

Intended to learn unlabelled texts from both left and right contexts, BERT is introduced in [9] and had advanced the state-of-the-art of eleven natural language processing tasks. Using the Masked Language Model (MLM) objective, the model randomly masks some tokens from the input and demands the model to predict the ground truth label of the masked token based on its context. During the pre-training phase, 15% of the input tokens will be uniformly chosen for masking, which is masked 80% of the time; for 10% of the time, the word is replaced with a random word; and for the remaining time, the word is kept unchanged [20]. MLM allows the representation to combine the left and right context, realising the training of a deep bidirectional Transformer. Moreover, a "next sentence prediction" task is also utilized to pre-train text-pair representations. To obtain the vectors of a text, the model takes in a token sequence, often the output of the accompanying tokenizer, and inserts special tokens such as `[CLS]` and `[SEP]`. These tokens allow the model to handle a single sentence and a pair of sentences in a single token sequence. After tokenization, the token sequence is then passed into Transformer blocks to generate token-type embeddings, segment-type embeddings and positional embeddings (see Figure 1), and these three embeddings are summed as the final embeddings. As the model is designed for transfer learning, custom heads (or "poolers" in some literature) can be added to the model to suit different purposes such as classification or regression. In the present project, we use the classification head which contains two linear layers.

3.2.2 DistilBERT

DistilBERT, as the name suggests, is the "distilled", or simplified, version of BERT. Introduced in [30], DistilBERT's number of parameters is only 60% of that of a BERT model while retaining 97% of its language understanding capabilities and being 60% faster. To achieve this, knowledge distillation is leveraged to compress a larger "teacher" model to a more compact "student" model, reproducing the teacher's behaviour. The training loss used in knowledge distillation is a linear combination of the distillation loss, the MLM loss and the cosine embedding loss. DistilBERT has the same general architecture as BERT; however, the token-type embeddings and the head are removed, and the number of layers is halved. To train the student DistilBERT model, it is initialized from the teacher model by taking one layer out of two thanks to the common dimensionality. With the triplet loss, the 40% smaller DistilBERT is proven to have a similar capability to the BERT model, and the performance on the IMDb dataset is only 0.6% behind BERT. Distillation is done by utilizing gradient accumulation and removing the "next sentence prediction" objective. By removing the computationally intensive layers, the model works 60% faster during inference time. Excluding the tokenization step, DistilBERT is 71% faster than BERT on iPhone 7 Plus. In short, the invention of DistilBERT allows developers to deploy smaller models without compromising the performance of the model.

3.2.3 RoBERTa

Conducted as a replication study, [20] discovered that BERT was significantly under-trained, and can match or outperform post-BERT models if the training method is improved. Thus, the Robustly Optimized BERT Approach (RoBERTa) is introduced. This recipe includes modifications such as training BERT longer with larger batches and datasets, scrapping the "next sentence prediction" objective, training on longer sequences and dynamically changing the masking pattern applied to the training data. The research team suggested using dynamic masking because the original BERT implementation generates the masking pattern only during data preprocessing, which may cause overfitting issues. Results show that by introducing dynamic masking, the performance is similar to static masking but with improved efficiency. Also, removing the next sentence prediction task allows the model to match BERT's performance, disproving the belief that the next sentence prediction task is a crucial part of BERT. Training with larger batches also reduces the perplexity of the prediction; in other words, the model can predict texts better. Overall, RoBERTa is better than BERT over eight natural language processing tasks by 1-2% points and achieved state-of-the-art on GLUE, RACE and SQuAD datasets, which demonstrates that BERT, after refining the design decisions, remains competitive even after the invention of newer models. In this project, we will use the distilled version of RoBERTa, DistilRoBERTa, to conduct experiments. Similar to DistilBERT, it is distilled from a RoBERTa model with half the layers and has 40% fewer parameters than RoBERTa.

3.3 Late Fusion Multimodal Model

As discussed in section 2, late fusion aggregates the predictions made by several unimodal models for generating the final decision through non-parametric methods such as voting and weighted sum, or more sophisticated architectures such as neural networks, which make looser and looser initial conditions on the contribution and dimensions of the modalities [16, 23, 25–27, 31, 37, 41]. In our project, the late fusion model is compiled by combining the DistilRoBERTa and ResNeXt models, the two models that achieved the best performance in the classification task to be discussed hereafter. We choose to simply take averages of the visual and textual modalities, assuming the contribution of the two modalities is equal and forcing the dimension of the two embeddings to be the same. To fuse the two modalities, we first feed the visual and textual inputs into the corresponding models to obtain an embedding of the two features. The two obtained embeddings are then averaged for each sample. The averaged result is then passed into the SoftMax function as in normal classification tasks or is returned as an embedding of the sample for further processing. By employing the late fusion model, we can make full use of the modalities we have and provide more accurate results.

3.4 Vision-and-Language Transformer

ViLT model was originally proposed as a minimal and efficient vision-and-language pre-training model for image captioning tasks [11]. It consists of an image embedder for visual embeddings, a text embedder for textual embeddings, and a Transformer encoder for learning the interaction between each modality embedding. Compared to traditional models that retrieve image grid features with a separate deep convolutional branch, ViLT generates visual embeddings efficiently with the linear projection of image patches. Given an image-caption input pair, the text inputs are tokenized and embedded with a BERT embedder, while the image inputs are partitioned into 32 * 32 patches and then linearly projected to the visual embedding space. The visual and textual embeddings are then merged respectively to their modality type to produce a learnable contextualized vector, which will be forwarded to the encoder for modality interaction and feature extraction. The encoder is instantiated from standard ViT (see section 3.1). The model architecture is illustrated in Figure 5.

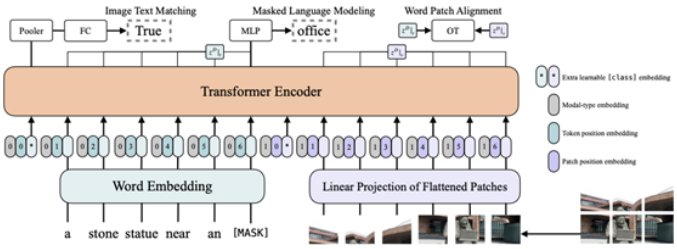


Figure 5: The model architecture of Vision–and-Language Transformers. Text and image patches are processed by the word embedding and linear projection layers respectively before forwarding to the Transformer encoder [11].

In order to precisely capture the relationships between visual and textual modalities, ViLT is additionally trained upon two different objectives: Masked Language Modeling (MLM) and Image-Text Matching (ITM). MLM randomly masks a portion of the inputted text tokens, requiring the model to predict their ground truth label based on the contextualized vector. On the other hand, ITM randomly assigns unaligned text captions to part of the images so that the model can distinguish the valid aligned pairs from the counterfeits. Intuitively, MLM allows the model to focus on 'how the image tokens can help determine the masked text tokens', while ITM requires the model to determine 'whether the input text accurately depicts its paired image'. The model can thus learn the cross-modalities interactions and adjust the contextualized vector accordingly with these objectives. To assure that the model is adapting to both textual and visual modalities but not relying solely on any of them, the technique of whole word masking is suggested for masking the entire word instead of just part of the word pieces. For example, given the word piece tokens `["em", "##bed", "##ding"]` generated from the word "embedding", either all or none of the tokens will be masked. With this simple strategy, the model cannot derive the masked tokens from its coupled

word pieces and must attend to the visual information for predicting the blanked word. RandAugment, an automated augmentation technique that transforms images with randomized operations of different magnitudes, is also used to augment the image inputs for the better generalization of vision models.

3.5 Contrastive Loss

Contrastive loss is a model loss function dated back to 2005. It was first proposed as a similarity metric for recognition and verification applications where categories are not known in advance and where the number of samples for a single category is small. [7] Unlike cross-entropy or Softmax loss, which train the model based on the predicted probabilities of each category, contrastive loss requires the model to minimize the embedding distance between similar samples *or* maximize that between samples in different categories, depending on whether the sampler chooses a positive or negative sample to make a pair. A sample pair contains the input sample as the "anchor" and a random sample from the whole dataset as the "positive" if the samples are in the same category or as the "negative" if they do not. As illustrated in Algorithm 1, to minimize the contrastive loss, it demands the model to minimize intra-class distance and maximize inter-class distance. To achieve the best performance, a pair miner selects either a "hard positive", the furthest positive sample, or a "hard negative", the nearest negative sample, to form the sample pair for loss propagation. Algorithm 1 illustrates the algorithm in pseudo-code format.

Algorithm 1 Contrastive Loss Algorithm

$margin = 1.0$
$loss = 0.0$

for X **in** SampleLoader **do**
  $IsPos$ = random(0, 1)
  $A =$ X ▷ Anchor
  $target_A =$ label of A
  if $IsPos = 1$ **then**
    $T =$ PairMiner(A) ▷ Positive
  else
    $T =$ PairMiner(A) ▷ Negative
  $target_T =$ label of T
  $emb_A, emb_T =$ Model(A, T)
  $dist =$ distance(emb_A, emb_T) ▷ Euclidean distance
  $loss = 1(target_T = target_A) \times dist^2$ + $1(target_T \neq target_A) \times (\max(margin - dist, 0.0))^2$
  ▷ Contrastive Loss Computation

return $loss$

3.6 Triplet Loss

Triplet loss is a model loss function commonly used in contrastive learning tasks. It was first proposed to learn useful representations by distance comparisons for classification tasks and was then popularized as a method to retrieve local feature descriptors in computer vision tasks [14, 34]. Similar to contrastive loss, triplet loss requires the model to minimize the embedding distance between similar samples while maximizing that between samples in different categories. A sample triplet consists of the input sample as the "anchor", the sample in the same category as the "positive", and the sample in different categories as the "negative". The sample triplets are forwarded to the model to generate anchor, positive, and negative embeddings. As illustrated in Algorithm 2, the minimization of triplet loss requires close anchor-positive distances and far anchor-negative distances. During training, a triplet miner is usually deployed to select "hard positive", which are the positive samples with the farthest distance from the anchor, and "hard negative", which are the negative samples with the closest distance from the anchor, to form the sample triplet for loss propagation.

Since no categorical probabilities are directly computed, triplet loss is suitable for tasks with large amounts of categories, such as person re-identification and speaker recognition. Besides, compared with the plain contrastive loss function, which only uses a positive sample or a negative sample for the loss computation, triplet loss considers both types of sample pairs at the same time, allowing the model to adjust the relative distances between an anchor and its counterparts more easily. In this project, we apply triplet loss to group similar product records for the product matching task. Algorithm 2 illustrates the triplet loss algorithm in pseudo-code format.

Algorithm 2 Triplet Loss Algorithm

$margin = 1.0$
$loss = 0.0$

for X **in** SampleLoader **do**
  $A = \text{X}$ ▷ Anchor
  $P, N = \text{TripletMiner}(A)$ ▷ Positive, Negative
  $emb_A, emb_P, emb_N = \text{Model}(A, P, N)$
  $l = margin + \text{distance}(emb_A, emb_P) - \text{distance}(emb_A, emb_N)$ ▷ Triplet Loss Computation
  $loss \mathrel{+}= \max(l,\ 0.0)$

return *loss*

3.7 Omni Retrieval

Omni Retrieval (Omni) is a pre-training objective for visual and textual information retrieval. It was proposed by the Facebook AI team to handle a massive amount of cross-modal and cross-pair e-commerce product data in their marketplace [39]. During training, the model first accepts product information from different sources to generate visual, textual, and multimodal embeddings. For any pair of products, their embeddings in the above modalities will be used for the contrastive loss function. The contrastive loss is applied over all 9 possible modality embedding pairs formed by the product pair, namely image-to-image, image-to-text, image-to-multimodal, text-to-image, text-to-text, text-to-multimodal, multimodal-to-image, multimodal-to-text, multimodal-to-multimodal. Since all sources of information are compared in the process, Omni is believed to foster embedding alignments across modalities, which could benefit the later downstream tasks over multimodal e-commerce products. In the ablation study conducted by the paper authors [39], models trained with Omni have shown significant improvements in several downstream tasks, especially in query-to-product retrieval, image-to-product retrieval, and image-to-product-image retrieval. Algorithm 3 outlines the training process of Omni Retrieval.

In the paper settings, Omni is adopted to pre-train product posting records without annotation data. In other words, the contrastive loss function inside Omni does not take in any labelling information as inputs. Since our datasets already provide the sample annotation data, we replace the Omni inner contrastive loss function with its variant that could consider the true sample labellings. To further improve the contrastive learning performance, we select triplet loss as the inner loss function so that both positive and negative samples can be compared at each training step. More details about the dataset experiments are discussed in Section 4.2.

Algorithm 3 Omni Retrieval with Triplet Loss

$margin = 1.0$
$loss\_fn = \text{TripletLoss}(margin)$
$loss = 0.0$
$omni\_feats = \{img, txt, mm\}$ ▷ Features used for Omni Retrieval

for X **in** SampleLoader **do**
  $A = \text{X}$ ▷ Anchor
  $P, N = \text{TripletMiner}(A)$ ▷ Positive, Negative
  $A_{img,txt,mm}, P_{img,txt,mm}, N_{img,txt,mm} = \text{Model}(A, P, N)$ ▷ Visual, Textual, multimodal features
  for $feat_1$ **in** $omni\_feats$ **do**
    for $feat_2$ **in** $omni\_feats$ **do**
      $l_{feat_1,feat_2} = loss\_fn(A_{feat_1}, P_{feat_2}, N_{feat_2})$
      $loss \mathrel{+}= \max(l_{feat_1,feat_2},\ 0.0)$

return $loss$

4 Experiment

4.1 Datasets

To compare the product embeddings generated by the unimodal and the multimodal learning networks, We evaluate our approaches on the Fashion Product Images (FPI) dataset, the Farfetch Fashion Outfit Challenge (FFOC) dataset, and the Shopee Price Match Guarantee dataset. The datasets are open-sourced and can be retrieved online without additional access requests. The FPI and FFOC datasets contain fashion product records attached with item images and respective metadata, while the Shopee dataset consists of posting records associated with the product thumbnails and titles. We use the product metadata for the textual model training, the product images for the visual models, and both modalities for the ViLT model. We select the FPI and the FFOC datasets for categorical learning, and the Shopee dataset for contrastive learning.

4.1.1 Categorical Learning

The Fashion Product Images (FPI) dataset consists of 44,441 fashion product images with the associated fashion style metadata. Each item image has a resolution of 480 * 640 pixels (width * height), while the product styles metadata are presented in raw JSON files and contain hundreds of data attributes such as `season`, `gender`, `productDisplayName`, `usage`, and `styleOptions`. We empirically select the `productDescriptors` as the textual input. Each product also includes 3 categorical variables in a hierarchical order, which are `masterCategory`, `subCategory`, and `articleType`. The 3 categorical variables are concatenated into a single category as the model prediction target. Figure 6 shows a sample

from the FPI dataset.

id: 10001

gender: Women

masterCategory: Apparel

subCategory: Bottomwear

articleType: Skirts

baseColour: Blue

season: Summer

brandName: Palm Tree

productDescriptors-description-value:
Wash Care Machine Wash separately in cold water Use mild detergent Do not Bleach Avoid Soaking Remove promptly Line Dry inside out in shade Warm iron only Do Not Iron Directly on Print/embroidery

Figure 6: A sample from the FPI dataset. Each sample consists of a product image and a list of product attributes.

Farfetch Fashion Outfit Challenge (FFOC) dataset consists of 398,670 fashion product records and 300,000 fashion outfits manually built by Farfetch stylists and fashion experts. The dataset is originally designed as a data challenge of the 2022 Special Interest Group on Information Retrieval Workshop on eCommerce [1]. Participants are required to develop a model that generates fashion outfits for each individual product based on its image and metadata. Since our work focuses on the generation of eCommerce product embeddings, the fashion outfit section of the dataset is neglected in this study. For the fashion product records, each item image is photographed with a frontal view on a white background and has a resolution of 255 * 341 pixels, while the product metadata contains 11 textual columns, namely `product_family`, `category`, `subcategory`, `gender`, `main_color`, `second_color`, `brand`, `materials`, `short_description`, `attributes` and `highlights`. While most of the columns are informative about the fashion style of the products, we empirically select `brand`, `short_description` and `highlights` as the textual input of the ViLT model. An ablation study is conducted to evaluate the importance of these textual columns in terms of product categorization performance. The fashion product category is produced by joining the values in the `product_family`, `category`, and `subcategory` columns. Figure 7 shows a sample from the FFOC dataset.

As aforementioned, the target categories in both datasets are composed of 3 different categorical variables. To ensure there are sufficient product samples for each target category, rare categories of the same parent category are merged as one while the categories without sufficient product records will be discarded. Different minimum sample thresholds are set based on the dataset size, namely 10 for the FPI dataset and 40 for the FFOC dataset. After preprocessing, the FPI and the FFOC datasets contain 121 and

Figure 7: A sample from the FFOC dataset. Each sample consists of a product image and a list of product attributes.

244 categories respectively. Figure 8 illustrates the target category distributions in terms of the primary category. As seen in the figure, clothing-related primary categories, such as Apparel (FPI) and Clothing (FFOC), contain more child categories and thus occupy larger portions of the dataset. To remove model biases towards frequent product categories, weighted samplers are used in the training step so that each product category has an equal chance to be selected. We then partition the datasets such that 80% of the data are used for training, 10% for validation, and 10% for testing.

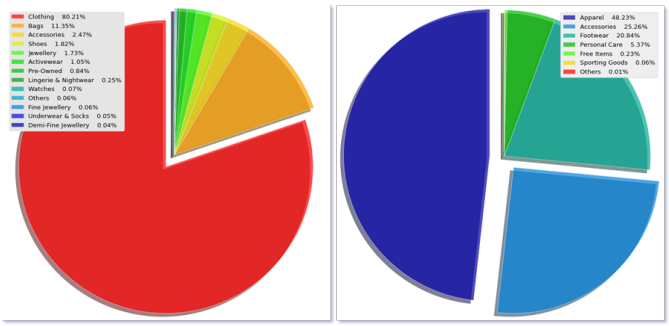


Figure 8: The primary category distribution of the FFOC (left) dataset and the FPI (right) dataset. Primary category refers to the `product_family` field in FFOC and the `master_Category` field in FPI.

4.1.2 Contrastive Learning

Removing near-duplicates is crucial for maintaining the content quality of large databases. In the context of online businesses, It is common to find the same product uploaded by different users with their versions of product images and descriptions. Without effective techniques to identify products posted repeatedly, retailers may fail to offer the same products at a more competitive rate compared with other competitors. The online e-commerce platform Shopee hosted the Price Match Guarantee competition on Kaggle to search for efficient solutions for the above challenge. The Shopee dataset consists of

34,250 e-commerce product records posted on the Shopee online shopping platform. Each row contains a product posting ID, a title description of the posting, and a product image uploaded by the retailers. Since Shopee is based in Southeast Asia, the product titles may consist of both English and Indonesian wordings. Each product posting is also associated with a single label group, an ID code that connects similar product posting records. There are 11,014 unique label groups, with each label group containing from 2 to 50 product postings. The main objective of the dataset is to return all product postings that are mapped to the label group of a given query product record. Figure 9 shows a sample from the Shopee dataset.

Due to the enormous amount of label groups, it is impractical to adopt the regular categorization task, which predicts the probability of each label group given a query product, as the model training objective. In real-life's cases, retailers are also expected to post products that do not relate closely to any of the previously-seen products, i.e. the new products should be assigned a new label group instead of the existing label groups. This challenge calls for the need for a contrastive learning objective, which projects similar product records in near embeddings while keeping the embeddings of dissimilar products distant from each other. The details of the contrastive learning process are discussed in Section 4.2.2. To fit the contrastive learning objective, we first identify all the unique label groups found in the dataset before partitioning them into the portion of 80%, 10%, and 10% for the training, validation, and testing split. All product records are then mapped to the respective split based on their label groups. This decision allows us to examine how well the model can generalize the concepts of product matching to new items with unseen label groups. As a result, there are 25,927 records of 8,811 label groups for training, 3,284 records of 1,101 label groups for validation, and 3,249 records of 1,102 label groups for testing. Unlike the case of categorical training, no weighted sampler is used during training with the Shopee dataset, since most product label groups have a limited sample size and we wish the model to learn from all available training records for a better generalization of the label group concept.

4.2 Training

4.2.1 Baseline models

For the visual baselines, we use a 50-layer ResNet model (ResNet50), a 50-layer ResNeXt model with 32 branches and a width of the bottleneck of 4 (ResNeXt50-32x4d) and adopt the ViT "Base" variant with a 16 * 16 input patch size (ViT-B/16), which are based on the original implementation in [10, 13, 36]. These models are scheduled using the `ReduceLROnPlateau` scheduler to reduce the learning rate when the validation loss has not decreased for three consecutive epochs.

For the textual models, we use the pre-trained DistilBERT base model, the BERT Base model and the

posting_id : train_104393727

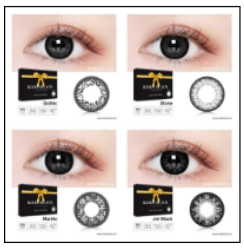
label group : 4292006141

title :
SOFTLENS KOKO JET BLACK MINUS
-0.50 s/d -5.00 / KOKO BLACK

posting_id : train_3804773252

label group : 4292006141

title :
SOFTLENS KOKO BLACK NORMAL
/ KOKO BLACK



posting_id : train_4167501582

label group : 4292006141

title :
Softlens KOKO BLACK NORMAL
& MINUS STONE JET BLACK
MARBLE GOTHIC
(-0.25 S/D -3.00) MINUS

Figure 9: Samples associated with the same label group in the Shopee dataset. Each sample consists of an item image and a product title.

DistilRoBERTa base model publicly accessible at the Hugging Face library [4]. The late fusion model is compiled using the ResNeXt model and the DistilRoBERTa model. All baseline models are trained for ten epochs using an AdamW optimizer with default settings, which has a learning rate of 1e-4 and a weight decay of 1e-2. We follow the best practices for training Transformer-based models, which refers to using a warm-up scheduler and then decaying the learning rate. For textual models, we train the models with no warm-up steps and linearly decay the learning rate from 1e-4 to 0. Late fusion models are also trained using textual model configurations.

During training, the images are preprocessed by first resizing and cropping them to 384 * 384, and then normalizing the RGB values by subtracting the mean (0.485, 0.456, 0.406) and the standard deviation (0.229, 0.224, 0.225) as suggested by [8]. For all texts, sentences are tokenized using the corresponding model and are truncated to 256 tokens and padded with padding tokens to match the length of the longest token sequence. Other hyperparameters are kept constant throughout the experiments, but the batch size differs among the models to account for the limited memory size. To ensure the models are achieving the best performance, we save the model whenever the validation loss is less than the best validation loss during the ten epochs. After the training, we load the best model and evaluate it by computing its accuracy

of classification on the test set.

4.2.2 ViLT

We develop the multimodal product embedding model based on the open-source ViLT implementation in PyTorch [19]. Following the paper settings, the ViLT model is constructed such that the `bert-base-uncased` tokenizer is used to encode the text inputs, while the Vision Transformer base of 32-pixel patch size (ViT-B/32) is used as the Transformer component of ViLT. The default optimizer configurations are also adopted, which are the AdamW optimizer with a base learning rate of 1e-4 and weight decay of 1e-2, and the polynomial decay scheduler with 25000 warm-up steps and end learning rate of 0. In the categorial learning section, the ViLT models are trained from scratch for 10 epochs. While in the contrastive learning section, we adopt the transfer learning approach to first initialize our models with the weight file that is pre-trained over four image-text-retrieval datasets using MLM and ITM for 200,000 steps [19], and then train the models for 100 epochs. This is because the product matching task is more complex than the catalogue categorization task due to the enormous difference in the number of unique sample labels and the number of samples per label between our datasets (see Section 4.1). All learning hyperparameters, except for batch size, are kept the same in all experiments. Due to the limited GPU memory, we select a batch size of 14 for the categorical learning datasets and a batch size of 12 for the contrastive learning datasets. During training, the shorter edge of product images is resized to 384 pixels while the longer edge is adjusted to maintain the aspect ratio. The overview of our approach is illustrated in Figure 1.

Different from the original paper settings in which MLM and ITM were adopted at the pre-training stage only, we apply these two objectives along with the categorization and the product matching objective to train the ViLT model over the product datasets. This decision allows us to assess how MLM and ITM affect the model performance directly. Whole word masking and RandAugment are also applied to facilitate the multimodal fusion process of product images and metadata. Ablation studies are conducted to explore the contribution of each metadata attribute and different sources of information. In the catalogue categorization task, different combinations of product metadata are selected as the textual inputs. While in the product matching task, different combinations of product modality embeddings are selected as the comparative features for Omni Retrieval (see Section 3.6). The feature ablation study is conducted on the FFOC dataset for the categorical learning part, and on the Shopee dataset for the contrastive learning part. On the other hand, to estimate the effects of MLM and ITM in fusing multimodal information, we also conduct experiments on these pre-training objectives using all the aforementioned datasets.

4.3 Evaluation

We use the FPI and FFOC datasets for the catalogue categorization task and the Shopee dataset for the product matching task. For catalogue categorization, where the model predicts the category of a product catalogue, accuracy is selected as the evaluation metric to follow the common practice for classification datasets. For product matching, where the model returns products most near to the input samples and above the similarity threshold, F1-score is chosen as the evaluation metric. This decision allows us to efficiently assess the model sensitivity against both false positive and false negative predictions, especially in the case when the number of positive matches (products in the same posting group as the input sample) is much smaller than that of negative matches (products in posting groups different from the input sample). In addition to the F1 score, Normalized Discount Cumulative Gain (NDCG) is also reported for contrastive learning model evaluations. NDCG is a widely used machine learning metric for ranking systems, which considers both the presence and the position of the related sample matches in the model outputs. To achieve good NDCG, not only should the model find all associated sample matches, but also place those matches in the top position of the output list. We adopt NDCG as an evaluation metric to analyze the performance of our model in returning related product matches with high rankings.

5 Results

5.1 Categorical Learning

Model	# Params (M)	Image	Brand	Description	Highlights	Accuracy (%)
ResNet50	24.0	■				61.20
ResNeXt50-32x4d	23.3	■				61.03
ViT-B/16	86.3	■				52.05
DistilBERT	66.0			■		69.34
BERT	110.0			■		64.56
DistilRoBERTa	82.0			■		69.77
Late Fusion	105.3	■		■		27.55
ViLT-B/32	113.2	■	■			61.08
		■		■		78.93
		■		■		76.96
		■		■	■	77.09
		■			■	67.67
		■	■		■	69.65
		■	■	■	■	79.35

Table 1: Model classification performances with different input features on the Farfetch Fashion Outfit Challenge (FFOC) dataset. Greyed regions are the selected input features. All models are trained with the basic classification objective only.

As recorded in Tables 1 and 2, the accuracy rates of the visual baselines are at least 10% points less than that of ViLT. On the other hand, textual models are performing significantly better than visual models by at least 7% points. This implies the product images are either noisy or not as descriptive as the text description. Still, only when the dataset is small can the textual models match the performance of the ViLT models; the performance gap widens to 7-12% if the dataset size becomes large, as in the case of FFOC. These imply, by including textual data, the ViLT model can find the correlation between each word and the image patches, and thus use both pieces of information to predict the product categories with higher credibility. In contrast, the visual models can only classify products according to the product's image without any description or details about the image, while the textual models can only get the context of the words from their neighbours but cannot distinguish the visual subtleties. Since products in the same category can have various designs, it can be said that the visual models cannot distinguish products only by their appearance. The same can be said for the texts: since products of similar designs (say they are only different in colour) may have similar sentence structures, the impact of the altered token may be diluted by the overall tokens, hindering the models' ability to distinguish them. From these two results, we can observe that the inclusion of text provides extra descriptions of the image, such as positions, shapes, colours and brand of the clothing; visual images strengthen the connotation of the texts, allowing the model to associate a visual image with

Dataset	Model	# Params (M)	CLS	ITM	MLM	Accuracy (%)
FPI	ResNet50	23.8	■			75.26
	ResNeXt50-32x4d	23.3	■			78.09
	ViT-B/16	86.2	■			64.36
	DistilBERT	66.0	■			91.57
	BERT	110.0	■			90.26
	DistilRoBERTa	82.0	■			90.73
	Late Fusion	105.3	■			34.87
	ViLT-B/32	113.0	■			91.53
		113.0	■	■		92.19
		137.1	■		■	92.01
FFOC	ViLT-B/32	113.2	■			79.35
		113.2	■	■		81.30
		137.3	■		■	80.03

Table 2: Model classification performances with different training objectives. Greyed regions are the applied training objectives. CLS: classification, ITM: image-text matching, MLM: masked language modelling with whole word masking. All textual fields of the datasets are used in the ViLT-B/32 training.

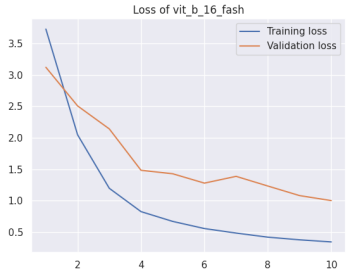
the word. This helps the model to distinguish products by combining knowledge from the two modalities.

Similarly, the late fusion model's performance is greatly hindered by the images' noisiness and the fusion method. First, using simple late fusion methods such as taking averages is a rather strict assumption on the modalities since the model is forced to consider the other modality. In reality, there is no need to make such an assumption. It is simply a design decision that we do not want the model to degenerate into a unimodal model, which defies our purpose for making a late fusion model in the first place. Second, such a simple method cannot associate the information from one modality to the other because each model is trained separately, which means that the combined model is not demanded to consider other modalities when making a decision. An improved model may attempt to use self-attention to generate a better embedding or at least use a weighted average of the two to allow the model to consider the modalities in different weights.

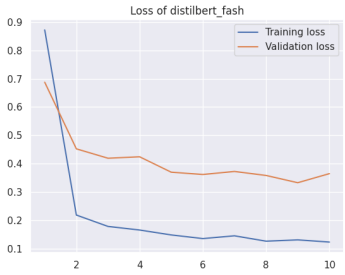
We also observed overfitting issues for all baseline models. For example, Figure 10 shows the six training curves from the visual baseline models on the FPI dataset. We can see that the model loss differs enormously during testing and validation. Using some image or text augmentation may alleviate the problem but, to our belief, the improvement in accuracy will be minimal and cannot overcome the performance gap. Concerning the magnitudes of the validation losses, we can observe the validation losses for textual models are always below 1.0 and the differences are always under 0.2, while the validation losses for visual models and the differences are always around 1.0. These are pieces of collateral evidence that the images are noisy and hamper the inference accuracy.

(a) ResNet50

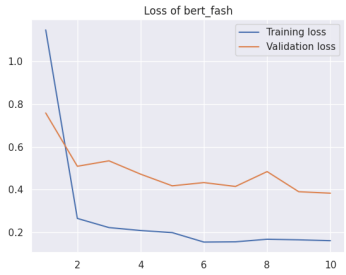
(b) ResNeXt50-32x4d



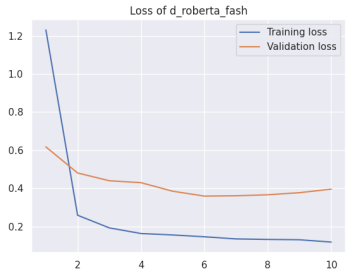
(c) ViT/B-16



(d) DistilBERT Base



(e) BERT Base



(f) DistilRoBERTa Base

Figure 10: Training curves of six models on FPI.

For the multimodal ViLT model, all the included text attributes contribute to the classification accuracy of the FFOC dataset, as illustrated in Table 1. The ViLT model achieves the highest classification accuracy of 79.35% when all text attributes are included in the textual input. The feature ablation study shows that the description field provides the most valuable product information compared with brand and highlights. Surprisingly, the product brand along with the description field yields a higher classification accuracy than the set-up for description and highlights, even though the training with only the brand field itself has relatively low accuracy. This could imply that the brand field carries the least classification-relevant information, but this information greatly complements the descriptive power of other textual fields. On the other hand, adding product highlights to the description field does not bring significant performance improvement. This could imply that the highlights and the description fields are highly similar, and thus concatenating these two fields will not provide much new information to the model learning process.

Table 2 shows the model classification performance after training with different objectives. After jointly training the ViLT model with MLM and ITM objectives, the model accuracy has shown improvements of 0.48% to 1.95% in both datasets. In terms of the training cost, the ITM head does not introduce many extra parameters to the model. In contrast, the MLM head costs an additional 24.1M parameters due to the new masked token prediction head of extensive vocabulary size. Though requiring more parameters and memories to train, the model with MLM objective does not perform better than its counterpart with ITM objective. MLM requires the model to predict the masked text tokens based on the visual information,

which theoretically helps align the visual and textual embedding. On the other hand, ITM instructs the model to learn the consistency between product images and their coupled metadata for effective integration of visual and textual information. Though both objectives are widely adopted in training vision and language datasets, for the FFOC dataset, the models trained with ITM perform slightly better than those trained with MLM. One possible reason for this performance gap is that random masking of the textual input by MLM may have disabled certain keywords that are crucial for the categorization task and thus hinders the aggregation of cross-modality product data. It is also believed that the vision source is a more informative modality than the textual source for our fashion product datasets, in the sense that the product metadata may lack information about its item appearance but contain attributes unrelated to its category. These unaligned image-description pairs lead to the limited benefits of using MLM as the training objective. On the other hand, by concerning the consistency across inputs of different modalities, ITM implicitly guides the model to sharpen and preserve distinction features from the product image and their metadata. Thus, though both objectives could improve the model accuracy, ITM is a more effective training objective than MLM in the fashion product categorization that emphasizes the visual information of the item.

5.2 Contrastive Learning

5.2.1 Contrastive loss

Model	# Params (M)	Contrastive Loss		Triplet Loss	
		F1 Score (%)	NDCG (%)	F1 Score (%)	NDCG (%)
ResNet50	24.0	41.50	63.81	50.94	62.55
ResNeXt50-32x4d	23.3	49.84	64.33	53.36	63.37
ViT-B/16	86.3	59.34	74.15	58.93	79.44
DistilBERT	66.0	53.02	79.03	54.69	85.75
BERT	110.0	52.07	69.16	52.04	82.12
DistilRoBERTa	82.0	18.37	55.84	54.71	80.59
Late Fusion	105.3	54.22	61.70	48.67	74.43

Table 3: Model performances with contrastive loss and triplet loss.

Table 3 shows the baseline models' results in the product matching task using contrastive loss and triplet loss. For contrastive loss, we can see that ViT is the best visual model and DistilBERT is the best textual model. Except for DistilRoberta, other models yield an F1 score of more than 50% and an NDCG of more than 60%. Transformer-based models such as ViT and BERT yield similar results. Surprisingly, even though the loss for DistilRoBERTa is similar to that of BERT and DistilBERT, the F1 score is significantly lower than the other two models. Comparing the results using contrastive loss and triplet loss, it can be observed that using triplet loss allows the model to produce better and more refined results. As discussed

in Section 3.6, using triplet loss causes a slight increase in F1 score and a significant increase in NDCG because we allow the model to compare two pairs of distances at the same time, which reduces the distance of similar products and increases the distance of dissimilar products at the same time. Comparing the learning curves of the two methods, we can see that the learning curves for triplet loss show less overfitting (see Appendix A, Figures 13 and 14). The comparison of baseline models and ViLT using triplet loss will be discussed below.

5.2.2 Triplet Loss

Model	# Params (M)	Omni Retrieval Feats			Extra Objectives		F1 score (%)	NDCG (%)
		Multimodal	Vision	Textual	ITM	MLM		
ResNet	24.0						50.94	62.55
ResNeXt	23.3						53.36	63.37
ViT	86.3						58.93	79.44
D-BERT	66.0						54.69	85.75
BERT	110.0						52.04	82.12
D-RoBERTa	82.0						54.71	80.59
Late Fusion	105.3						48.67	74.43
ViLT	111.0						56.61*	77.43*
	111.0						78.83	93.92
	111.5						66.67	87.02
	111.5						78.34	94.12
	112.0						74.92	91.68
	135.1						72.94	91.81
							80.44	95.47
							73.99	92.91
	136.1						77.41	93.20
							79.29	94.27
							79.54	94.43

Table 4: Model performance with triplet loss. ResNet, ResNeXt and ViT are visual models while DistilBERT, BERT and DistilRoBERTa are textual models. All ViLT model runs, except for the one marked with an asterisk, are initialized with the pre-trained model weights.

Table 4 shows the baseline models' results in the product matching task using triplet loss. Except for ViT, the F1 scores of the baseline models are all around 50%, implying that models are fairly accurate in identifying similar products relying on either the images or the description of the products. Among these models, we can see that ViT is the best visual model and DistilBERT is the best textual model. Although the textual models yield similar F1 scores, DistilBERT performs significantly better in returning more similar products on higher rankings. Moreover, the performance of the late fusion model is also worth noting. Although the

model's performance on classification is only around 30%, the averaged embeddings yield similar predictive power and are even better in providing more similar results than visual models, which implies that the use of two or more modalities, even combined very simplistically, helps refine the prediction results. Still, only a few baseline models can outperform the train-from-scratch ViLT model on NDCG, and none of them outperforms the pre-trained ViLT models on either indicator. Comparing the indicators, we can see that ViLT produces more accurate and quality results by returning more similar products and ranking them higher.

For the multimodal ViLT model, we trained the model using the triplet loss version of Omni Retrieval, as illustrated in Sections 3.5 and 3.6. As shown in the table, models initialized with the pre-trained checkpoint have shown significant F1 score improvement of 10.06% to 23.83%, compared with the train-from-scratch baseline model that only attains around 56% of the score. Given that the models perform fairly well for the categorical learning part, this result could imply that the product-matching task requires much more previous knowledge than the catalogue categorization task, especially when the number of product posting groups is around 50 times or more than the number of product categories (see Section 4.1). Besides, it is also noticed that including more modality pairs for Omni Retrieval does not necessarily lead to better model performances. In our case, the model attains the highest F1 score of 80.44% and NDCG of 95.47% when only the product multimodal embeddings are used for Omni Retrieval, that is, a single triplet loss term computed from the multimodal embeddings of product pairs. The model performance degrades slightly for 0.49% after including textual embeddings for Omni Retrieval, and degrades significantly for 12.16% when vision embeddings are included. A possible explanation could be the information irrelevance between the image and the text parts of product samples. For example, the image of the item "SoundWave Pro Wireless Headset", instead of showing the appearance of the actual item, may display mainly text banners about the product specifications, like "Last for 24 Hours", "Support Wireless Charging", "Water-Proof" etc. In addition, some types of products may also share similar item images as they have identical packaging except for the product title printed on top of it, such as beverage cartons and phone cases. The vision encoder only perceives the product appearances and hardly captures these printed texts on the item packaging without optical character recognition. Under Omni Retrieval, the model is required to generate similar embeddings for these common product packaging and distinctive features from other modalities. Producing aligned cross-modality embeddings for these irrelevant product image-description pairs may introduce more noise to the learning process, thus deteriorating the model product matching performance.

Apart from the modalities used, we also evaluate the effect of ITM and MLM on the training performance of Omni Retrieval using the multimodal-embedding-only setting and the all-modality setting. We train the ViLT models with Omni Retrieval loss and the above extra learning objectives concurrently. As shown in Table 3, the F1 score performances of both settings have improved by 1.61% and 4.37% respectively after training with the MLM objective. This result could imply that MLM is effective in improving the quality

of multimodal embeddings of both training settings. However, for the ITM objective, the F1 score of the multimodal-embedding-only Omni Retrieval setting has declined seriously by 5.89%, while that of the all-modality setting has improved by 2.49%. Based on this performance gap, it can be seen that ITM helps in aligning the textual and visual embeddings, but in the meantime may degrade the high-dimensional projections of multimodal features. While this explanation seems to disagree with the conclusion drawn in the categorical learning section, it has to be noted that the importance of visual inputs could vary greatly with our choice of product datasets in this section. Unlike general e-commerce products, the concept of a particular fashion product usually relates to its item appearance shown in the thumbnail. For example, the concept of a tropical hat can be deduced from the product style and shape inside thumbnails, but it could be difficult to identify a particular product as cold medicine just by reviewing its item appearance without any description. The alignment between visual and textual information for multimodal fusion, which the ITM objective aims to promote, does not always apply to e-commerce data in real life. Thus, multimodal embeddings learned from the ITM objective may not be suitable for e-commerce product datasets. Still, ITM is still effective in improving the textual and visual embedding themselves, as supported by the evaluation results of the all-modality setting. After training with both ITM and MLM objectives concurrently, the all-modality setting has shown an F1 score improvement of 4.62%, while the F1 score of the multimodal-embedding-only setting has declined by 4.84%. Such performance degradation in the latter setting is likely due to the presence of the ITM objective, as previously discussed. To summarize, though both MLM and ITM are commonly used in pre-training large language models, MLM can assist our ViLT models in aggregating cross-modality information and yield more representative multimodal embeddings for product matching, while the effectiveness of ITM depends on the alignment between visual and textual inputs, which can hardly be guaranteed in our e-commerce product dataset.

5.3 Visualizations

We produced the training curves in Appendix A to better explain the difference in performance between the unimodal and multimodal models. To visualize the resultant unimodal embeddings, we utilize TensorBoard, a visualization tool developed by the TensorFlow team. With the collaboration of PyTorch and TensorFlow, PyTorch has provided an easy application programming interface to visualize PyTorch tensors. The visualizations and an in-depth analysis of the results can be found in Appendix B.

6 RoBERTa and XLM-RoBERTa

As previously noticed in the last term report, we would use skip-gram and other advanced text models such as RoBERTa and XLM-RoBERTa. However, we scrapped these plans after implementing them for training. For the skip-gram model, it has been ten years since its invention and more advanced transformer

models such as BERT had already been the mainstream in the field of natural language processing; therefore, there is no reason to persist in using skip-gram. Moreover, the results are very poor (about 1%) on all tasks, which does not help clarify the importance of using multimodal learning. For RoBERTa and XLM-RoBERTa, we scrapped them since the results are also under 1%, which is out of our expectations.

We believe there are no flaws in the original implementations of RoBERTa and XLM-RoBERTa since the models are adopted from Hugging Face and the training scripts for the textual models are basically a clone, with necessary modifications, of those for visual models. Thus, we suspect the number of parameters may contribute to poor performance due to underfitting. RoBERTa and XLM-RoBERTa have 125M parameters, while the models we finally adopt all have a number of parameters not more than 110M. As discussed in Section **??**, the distilled versions of BERT and RoBERTa have half the layers of the original without significantly hampering the performance of the models. Therefore, using a model too large can exacerbate the training performance.

One may also suggest that the reason is that the models are using different tokenization algorithms. In Hugging Face's implementation, BERT uses WordPiece tokenization while RoBERTa uses Byte-Pair Encoding (BPE) tokenization, which differs from how the words are being tokenized. [2, 6] To be concise, WordPiece and BPE both decompose words into letters and then attempt to find merge rules; however, BPE extracts the most frequent word pair as the merge rule while WordPiece calculates a score using their frequency and extracts the pairs with the highest score. [6] However, that does not seem to explain the discrepancy since DistilRoBERTa is also using BPE tokenization and has obtained satisfactory results. We suspect the reason lies in the architecture of the models, which requires much more effort for future investigation.

7 Conclusions

We successfully proved our postulate that blending two modalities yields better embeddings and representations by showing that the multimodal ViLT model matches or outperforms the baseline models on both catalogue categorization and product matching tasks. Supported by the ablation study on cross-modality features, we demonstrated that the ViLT model can combine information from both item images and product metadata, instead of solely relying on a single modality, to generate more representative embeddings that favour e-commerce downstream tasks. By including ITM and MLM as extra train objectives, our multimodal ViLT models achieve higher performance metrics in several downstream tasks, as shown in the Results section. Though it has been seen in Section 5.2 that the nature and the quality of dataset samples are great factors in the model training effectiveness, we strongly believe that multimodal learning, with careful tuning and selection of learning

objectives, could yield more comprehensive product embeddings for general e-commerce downstream tasks.

While the result is promising, the project has challenges and limitations. One is the limited GPU resources. During our training process, programs often failed because the GPU did not have enough memory to store the batch of data and the model altogether, and we needed to reduce the batch size accordingly. Considering that our resources are limited and shared among other project teams, we cannot utilize deeper or larger models. The same can also be said of the input image size (384 * 384), which is already the largest possible size for the batches and model architectures to be stored in the memory. Apart from the hardware limitations, the sizes of the product datasets used in this project are also small compared with those used in real-life applications. None of the product datasets in this project has more than 0.5M samples, while it is common to use millions of product samples in model training for commercial use cases. For example, the Facebook AI research team collected 50M product catalogue posts and 52M marketplace posts to pre-train a large-scale commerce multimodal model [39]. Larger product datasets increase the number of product catalogues and the number of samples per catalogue, thus enabling the model to learn a more diverse and granular understanding of e-commerce contexts. Due to limited resources and time, we cannot afford larger datasets for training our models under several experimental settings. The lack of training samples limits the ability of our models to generalize into other e-commerce downstream tasks. Still, if time allows, we are convinced that the model performance can be further improved after training with more product samples.

As discussed in Section 5.1, the effectiveness of Omni Retrieval greatly depends on the relevance of cross-modality information. Increasing the number of samples per product catalogue may help to tackle this issue, as it prevents the model biases towards certain samples with irrelevant modalities. Still, future studies could investigate how to detect and handle irrelevant information from different modalities during model training. A potential research direction could be to develop learning objectives more robust to irrelevant product modalities, such as Omni Retrieval integrated with the optical character recognition module to identify descriptions printed on the product packaging. While products with irrelevant information from different modalities are inevitable in real life, certain procedures can be developed to filter out these noisy samples in the construction of product datasets for training. We also suggest further modifications to the multimodal learning networks, such as incorporating the ViLT model with other Transformer variants like PoolFormer [40] and Xformer [17], in pursuit of lighter and more efficient machine learning models that fit e-commerce practical use cases.

Acknowledgements

We thank our project supervisor Dr Lam King-tin for his generous support and practical suggestions. We also thank HKTVMall for authorizing the use of the Open Databank to support our exploration.

References

[1] The 2022 sigir workshop on ecommerce. URL: https://sigir-ecom.github.io/.

[2] Byte-pair encoding tokenization. URL: https://huggingface.co/course/chapter6/5?fw=pt.

[3] Embeddings | machine learning | google developers. URL: https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture.

[4] Hugging face. URL: https://huggingface.co/.

[5] Modality - definition, meaning & synonyms. URL: https://www.vocabulary.com/dictionary/modality.

[6] Wordpiece tokenization. URL: https://huggingface.co/course/chapter6/6?fw=pt.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. doi:10.1109/CVPR.2005.202.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi:10.1109/CVPR.2009.5206848.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL: http://arxiv.org/abs/1810.04805, arXiv:1810.04805.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL: https://arxiv.org/abs/2010.11929, doi:10.48550/ARXIV.2010.11929.

[11] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, Zicheng Liu, and Michael Zeng. An empirical study of training end-to-end vision-and-language transformers, 2021. URL: https://arxiv.org/abs/2111.02387, doi:10.48550/ARXIV.2111.02387.

[12] Ahmed Fathalla, Ahmad Salah, Kenli Li, Keqin Li, and Piccialli Francesco. Deep end-to-end learning for price prediction of second-hand items. *Knowledge and Information Systems*, 62(12):4541–4568, 2020. doi:10.1007/s10115-020-01495-8.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL: https://arxiv.org/abs/1512.03385, doi:10.48550/ARXIV.1512.03385.

[14] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network, 2018. arXiv:1412.6622.

[15] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Kazuhiro Sumi, John R. Hershey, and Tim K. Marks. Attention-based multimodal fusion for video description, 2017. URL: https://arxiv.org/abs/1701.03126, doi:10.48550/ARXIV.1701.03126.

[16] Shih-Cheng Huang, Anuj Pareek, Saeed Seyyedi, Imon Banerjee, and Matthew P Lungren. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *NPJ digital medicine*, 3:136, 2020. URL: https://europepmc.org/articles/PMC7567861, doi:10.1038/s41746-020-00341-z.

[17] Pranav Jeevan and Amit Sethi. Vision xformers: Efficient attention for image classification, 2021. URL: https://arxiv.org/abs/2107.02239, doi:10.48550/ARXIV.2107.02239.

[18] D. Kiela, E. Grave, A. Joulin, and T. Mikolov. Efficient large-scale multi-modal classification, 2018. URL: https://arxiv.org/abs/1802.02892, doi:10.48550/ARXIV.1802.02892.

[19] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision, 2021. URL: https://arxiv.org/abs/2102.03334, doi:10.48550/ARXIV.2102.03334.

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL: http://arxiv.org/abs/1907.11692, arXiv:1907.11692.

[21] Szymon Łukasik, Andrzej Michałowski, Piotr A. Kowalski, and Amir H. Gandomi. Text-based product matching with incomplete and inconsistent items descriptions. In Maciej Paszynski, Dieter Kranzlmüller, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot, editors, *Computational Science – ICCS 2021*, pages 92–103, Cham, 2021. Springer International Publishing.

[22] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining, 2018. URL: https://arxiv.org/abs/1805.00932, doi:10.48550/ARXIV.1805.00932.

[23] Emilie Morvant, Amaury Habrard, and Stéphane Ayache. Majority vote of diverse classifiers for late fusion. In Pasi Fränti, Gavin Brown, Marco Loog, Francisco Escolano, and Marcello Pelillo, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 153–162, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[24] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2539–2544, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL: https://aclanthology.org/D15-1303, doi:10.18653/v1/D15-1303.

[25] Shangran Qiu, Gary H. Chang, Marcello Panagia, Deepa M. Gopal, Rhoda Au, and Vijaya B. Kolachalama. Fusion of deep learning models of mri scans, mini–mental state examination, and logical memory test enhances diagnosis of mild cognitive impairment. *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, 10(1):737–749, 2018. doi:10.1016/j.dadm.2018.08.013.

[26] Dhanesh Ramachandram and Graham W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017. doi:10.1109/MSP.2017.2738401.

[27] Islam Reda, Ashraf Khalil, Mohammed Elmogy, Ahmed Abou El-Fetouh, Ahmed Shalaby, Mohamed Abou El-Ghar, Adel Elmaghraby, Mohammed Ghazal, and Ayman El-Baz. Deep learning role in early diagnosis of prostate cancer. *Technology in Cancer Research & Treatment*, 17:153303461877553, 2018. doi:10.1177/1533034618775530.

[28] A. Rubio, LongLong Yu, E. Simo-Serra, and F. Moreno-Noguer. Multi-modal joint embedding for fashion product retrieval. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 400–404, 2017. doi:10.1109/ICIP.2017.8296311.

[29] Gil Sadeh, Lior Fritz, Gabi Shalev, and Eduard Oks. Joint visual-textual embedding for multimodal style search. *CoRR*, abs/1906.06620, 2019. URL: http://arxiv.org/abs/1906.06620, arXiv:1906.06620.

[30] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL: http://arxiv.org/abs/1910.01108, arXiv:1910.01108.

[31] Ekaterina Shutova, Douwe Kiela, and Jean Maillard. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170, San Diego, California, June 2016. Association for Computational Linguistics. URL: https://aclanthology.org/N16-1020, doi:10.18653/v1/N16-1020.

[32] Yina Tang, Fedor Borisyuk, Siddarth Malreddy, Yixuan Li, Yiqun Liu, and Sergey Kirshner. Msuru: Large scale e-commerce image classification with weakly supervised search data. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019. doi:10.1145/3292500.3330696.

[33] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec. *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016. doi:10.1145/2959100.2959160.

[34] Daniel Ponsa Vassileios Balntas, Edgar Riba and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 119.1–119.11. BMVA Press, September 2016. URL: https://dx.doi.org/10.5244/C.30.119, doi:10.5244/C.30.119.

[35] Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. Deep multimodal fusion by channel exchanging, 2020. URL: https://arxiv.org/abs/2011.05005, doi:10.48550/ARXIV.2011.05005.

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2016. URL: https://arxiv.org/abs/1611.05431, doi:10.48550/ARXIV.1611.05431.

[37] Youngjin Yoo, Lisa Y. W. Tang, David K. B. Li, Luanne Metz, Shannon Kolind, Anthony L. Traboulsee, and Roger C. Tam. Deep learning of brain lesion patterns and user-defined clinical and mri features for predicting conversion to multiple sclerosis from clinically isolated syndrome. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 7(3):250–259, 2019. arXiv:https://doi.org/10.1080/21681163.2017.1356750, doi:10.1080/21681163.2017.1356750.

[38] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. URL: https://arxiv.org/abs/2205.01917, doi:10.48550/ARXIV.2205.01917.

[39] Licheng Yu, Jun Chen, Animesh Sinha, Mengjiao MJ Wang, Hugo Chen, Tamara L. Berg, and Ning Zhang. Commercemm: Large-scale commerce multimodal representation learning with omni retrieval, 2022. URL: https://arxiv.org/abs/2202.07247, doi:10.48550/ARXIV.2202.07247.

[40] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision, 2021. URL: https://arxiv.org/abs/2111.11418, doi:10.48550/ARXIV.2111.11418.

[41] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1114, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL: https://aclanthology.org/D17-1115, doi:10.18653/v1/D17-1115.

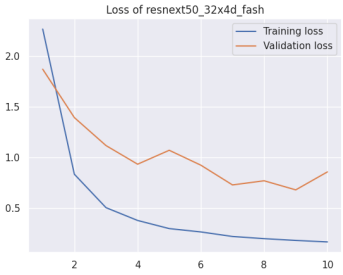
A Training curves

We produce our graphs of training loss and validation loss below.

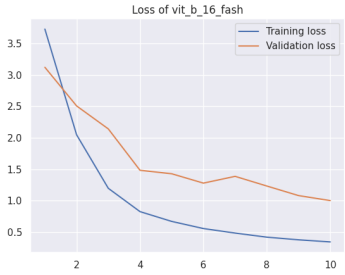
A.1 Baseline models

Figures 11, 12, 13 and 14 show the training and validation loss on FPI and FFOC classification tasks and the Shopee product matching task.

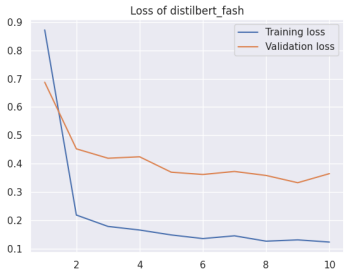
(a) ResNet50



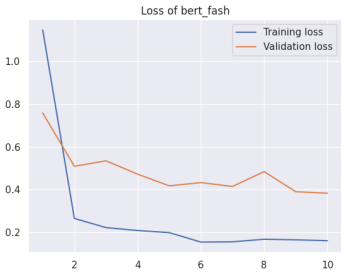
(b) ResNeXt50-32x4d



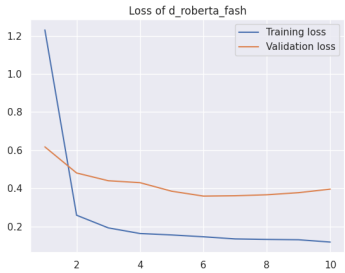
(c) ViT/B-16



(d) DistilBERT Base



(e) BERT Base



(f) DistilRoBERTa Base

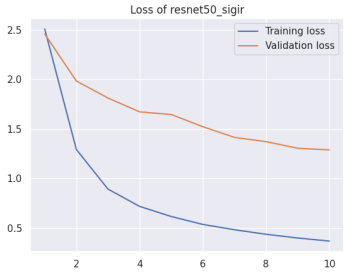
Figure 11: Training curves of six models on FPI.

A.2 Late fusion model

Figure 15 shows the late fusion model's training and validation loss on FPI, FFOC, and triplet loss.

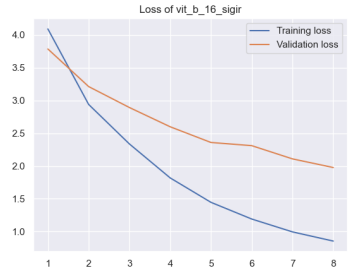
A.3 ViLT

Figure 16 shows the learning curves of the ViLT model on the Shopee Product Matching dataset under different Omni Retrieval experimental settings. More details can be found in Section 5.2.

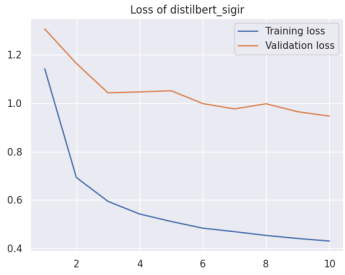


(a) ResNet50

(b) ResNeXt50-32x4d



(c) ViT/B-16



(d) DistilBERT Base

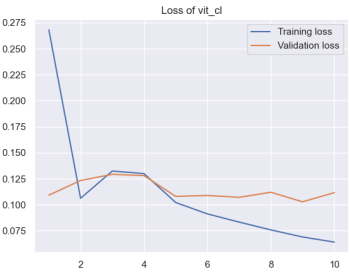
(e) BERT Base

(f) DistilRoBERTa Base

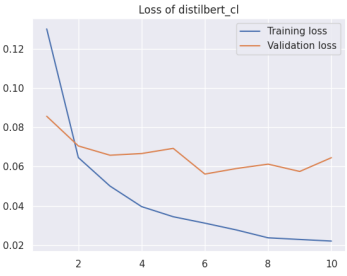
Figure 12: Training curves of six models on FFOC.

(a) ResNet50

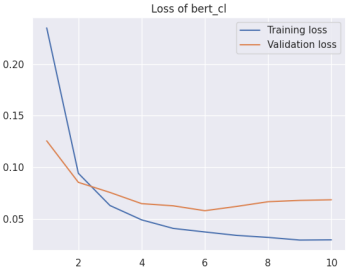
(b) ResNeXt50-32x4d



(c) ViT/B-16



(d) DistilBERT Base

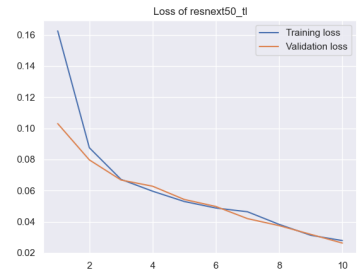


(e) BERT Base

(f) DistilRoBERTa Base

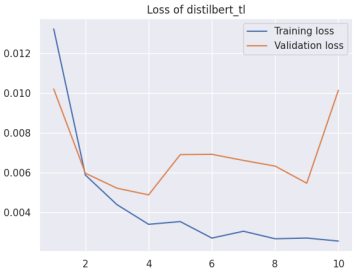
Figure 13: Training curves of six models on contrastive learning using contrastive loss.

(a) ResNet50

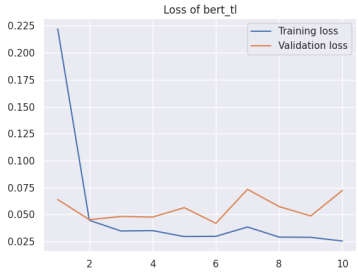


(b) ResNeXt50-32x4d

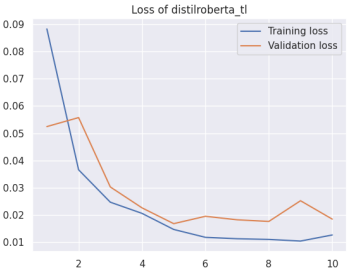
(c) ViT/B-16



(d) DistilBERT Base

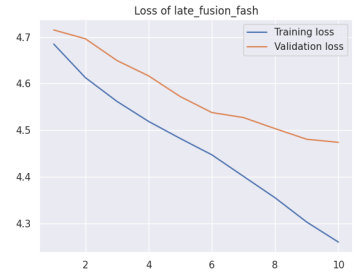


(e) BERT Base



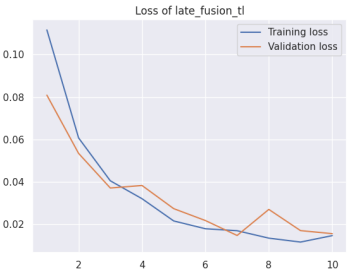
(f) DistilRoBERTa Base

Figure 14: Training curves of six models on contrastive learning using triplet loss.



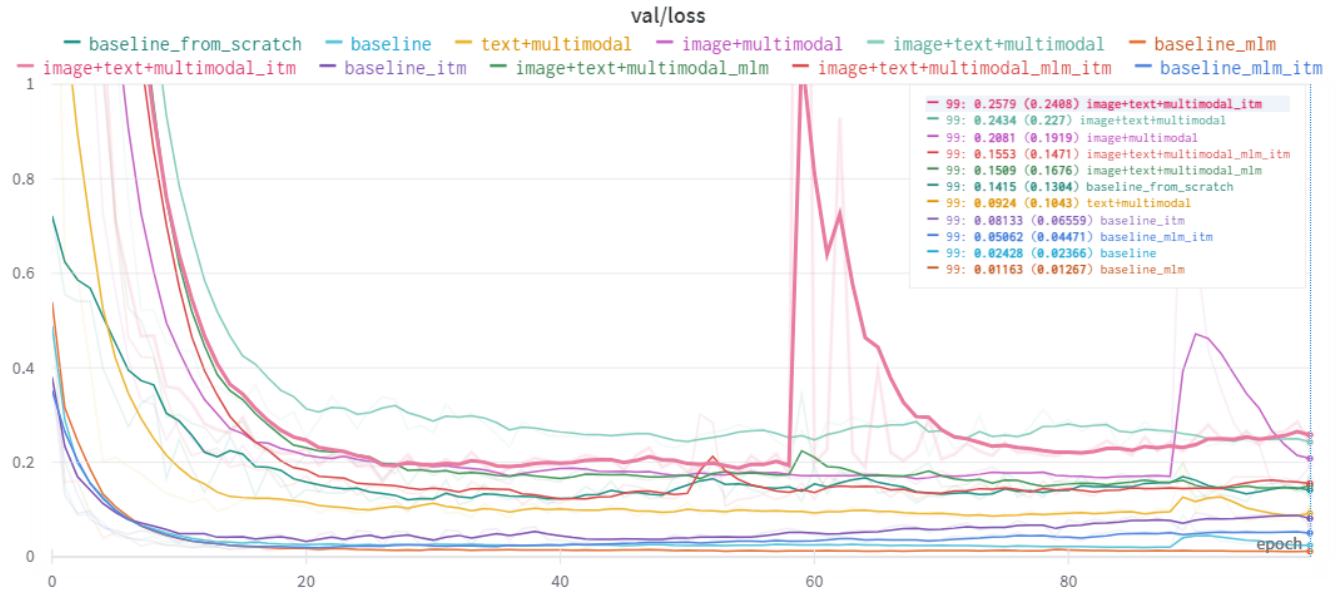
(a) FPI

(b) FFOC

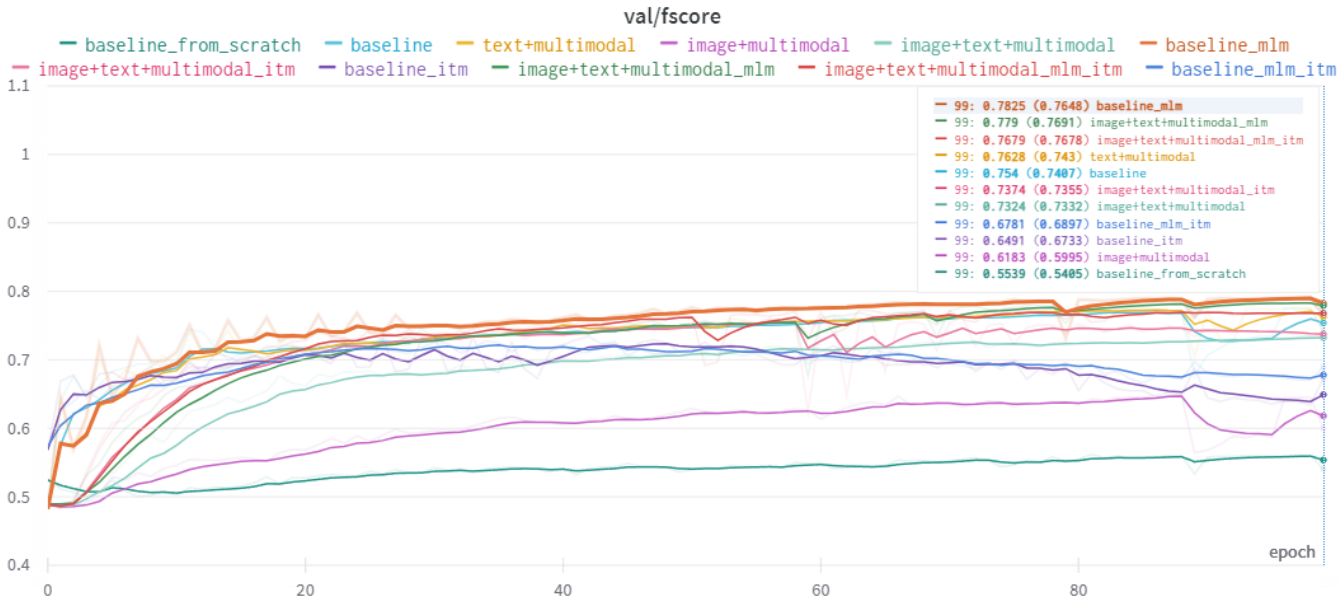


(c) Shopee product matching with triplet loss

Figure 15: Training curves of late fusion models on different tasks.



(a) Validation loss curve

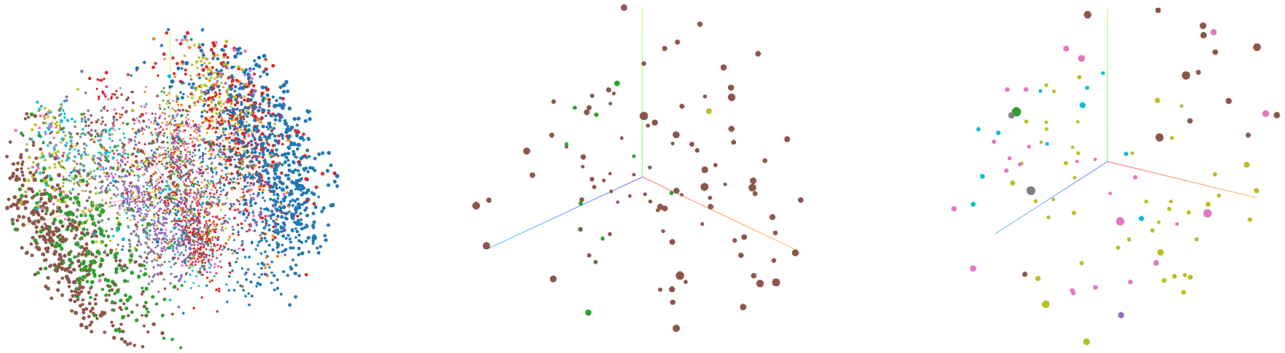


(b) Validation F1 score curve

Figure 16: Learning curves of the ViLT model on the Shopee Product Matching dataset under different experimental settings. For clearer visualization, the Y-axis value ranges have been adjusted and the metric curves have been smoothed by an exponential moving average in the scale of 0.5.

B Visualizations for unimodal models

We produce a visualization of unimodal embeddings in Figures 17, 18, 19, 20 and 21. We only provide the embeddings of the best-performing models for simplicity.

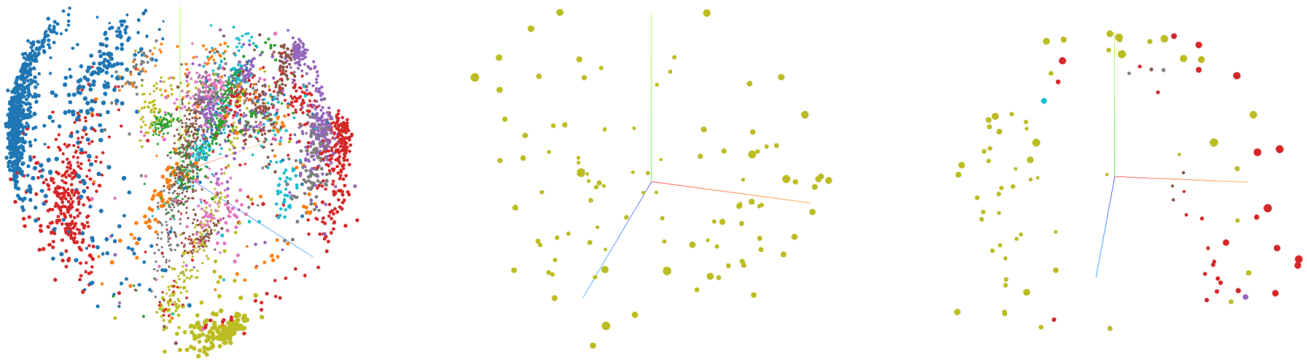


(a) Overview

(b) An example of good classification performance

(c) An example of bad classification performance

Figure 17: ResNeXt model embeddings for the FPI dataset.

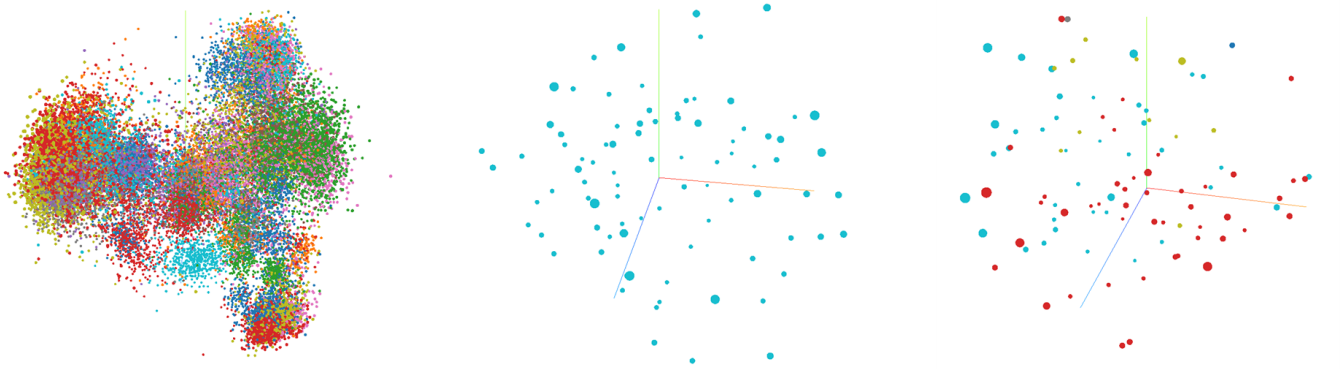


(a) Overview

(b) An example of good classification performance

(c) An example of bad classification performance

Figure 18: DistilBERT model embeddings for the FPI dataset.

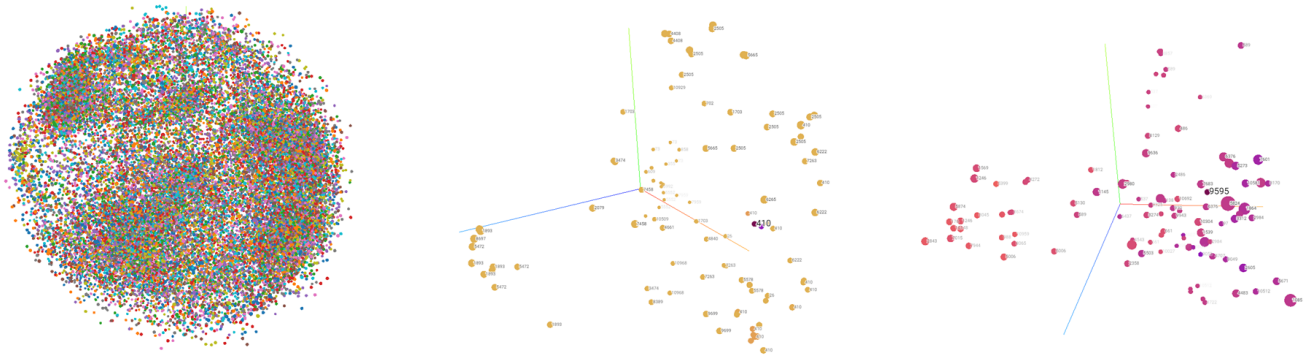


(a) Overview

(b) An example of good classification performance

(c) An example of bad classification performance

Figure 19: DistilRoBERTa model embeddings for the FFOC dataset.

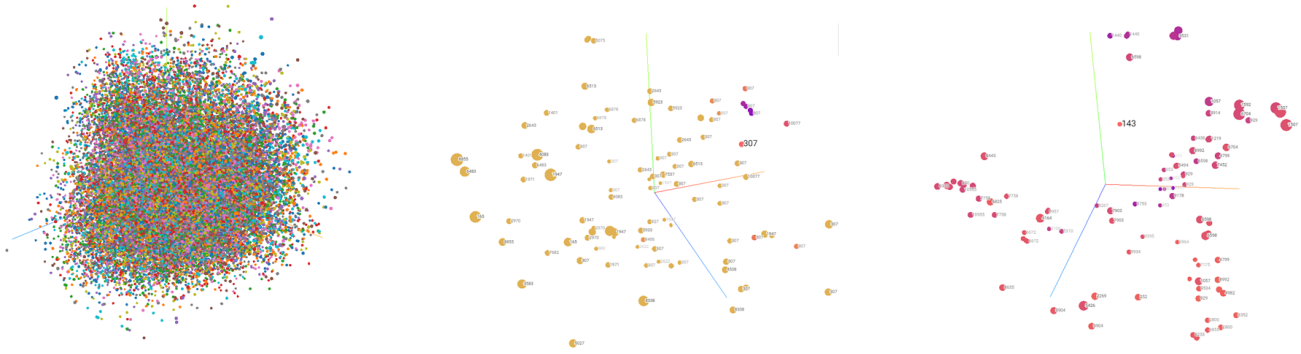


(a) Overview

(b) An example of good matching performance

(c) An example of bad matching performance

Figure 20: DistilRoBERTa model embeddings for the product matching task (contrastive loss).



(a) Overview

(b) An example of good matching performance

(c) An example of bad matching performance

Figure 21: DistilRoBERTa model embeddings for the product matching task (triplet loss).

C Division of Labour

Table 5 shows the contribution of each member to this project. Overall, Wing-Kit Chan is responsible for all the sections related to multimodal learning, while Chun-Kiu Kwok is responsible for all the sections related to unimodal learning.

Section	Wing-Kit Chan	Chun-Kiu Kwok
0. Abstract	✓	✓
1. Introduction	✓	✓
2. Related Work		
2.1 ↪ Image-based and Text-based Approaches		✓
2.2 ↪ Multimodal Fusion	✓	
3. Method		
3.1 ↪ Image-based Models		✓
3.2 ↪ Text-based Models		✓
3.3 ↪ Late Fusion Multimodal Model		✓
3.4 ↪ Vision-and-Language Transformer	✓	
3.5 ↪ Contrastive Loss		✓
3.6 ↪ Triplet Loss	✓	
3.7 ↪ Omni Retrieval	✓	
4. Experiment		
4.1 ↪ Datasets	✓	
4.2 ↪ Training	✓	✓
4.3 ↪ Evaluation	✓	
5. Results		
5.1 ↪ Categorical Learning	✓	✓
5.2 ↪ Contrastive Learning	✓	✓
5.3 ↪ Visualizations	✓	✓
6. RoBERTa and XLM-RoBERTa		✓
7. Conclusions	✓	✓
A. Training Curves	✓	✓
B. Visualizations for unimodal models		✓

Table 5: Labour distribution for this project.