Early Fusion with Contrastive Learning: A Lightweight Alternative for Multi-modal Classification

Felix Wernlein[1], **Abhik Jana**[2], **Sandipan Sikdar**[1,3]
Leibniz University Hannover[1], Indian Institute of Technology Bhubaneswar[2], L3S Research center[3]
felix.wernlein@stud.uni-hannover.de, abhikjana@iitbbs.ac.in, sandipan.sikdar@l3s.de

Abstract

With the emergence of numerous modalities, such as text, image, audio, etc., the use of effective multimodal systems has increased significantly. However, one of the significant challenges faced by such multimodal systems is effectively aligning and integrating diverse modalities. Several models have been proposed to address these issues; however, state-of-the-art performance is achieved by complex, heavyweight models (complexity measured in terms of trainable parameters) alone. Hence, we propose a simple yet effective lightweight framework explicitly designed for multimodal classification tasks, utilising the early fusion method combined with a contrastive learning approach. The early fusion method focuses on fusing different modalities at the input level, whereas contrastive learning allows a single modality to capture intra-modality relationships. Experiments on three different genres of multimodal classification datasets demonstrate that the proposed lightweight framework achieves performance comparable to the most competitive heavyweight state-of-the-art models and, in some cases, even outperforms them.

Keywords: Contrastive Learning, Early Fusion, Multi-modality

1. Introduction

With the increasing availability of different modalities such as texts, images, audio, and more, there is a growing need to develop effective, easy-to-use multimodal systems that can effectively utilize information from all modalities. Multimodal systems can leverage diverse data types to enhance their capabilities, demonstrating superior performance over their unimodal counterparts in tasks such as classification (Kiela et al., 2018; Wang et al., 2020) or sentiment analysis (Zadeh et al., 2017; Hazarika et al., 2020). For all these tasks, effectively fusing different modalities to capture inter-modality relationships remains a key challenge. Attempts have been made to fuse different modalities into a single feature using early fusion (Gadzicki et al., 2020a), late fusion (Snoek et al., 2005; Shi et al., 2021), Graph-based fusion (Zhao et al., 2024), and other methods. On the other hand, models like **MuGNet** (Lu et al., 2023) use an Attention-Based fusion module. In another work, Erickson et al. (2022) proposes **AutoGluon**, an ensemble-learning model for multimodal classification and regression tasks.

However, most of these state-of-the-art models are complex (with millions of trainable parameters) and, hence, computationally intensive. Therefore, in this paper, we propose a lightweight framework (Figure 1) for multimodal classification tasks using early fusion and contrastive learning. Experiments on three genres of multimodal classification datasets show that our proposed lightweight framework (early fusion has $\sim 1M$ parameters compared to **AutoGluon** with $355M$ trainable parameters) either outperforms the most competitive baselines or produces comparable performances. Note that, in this work, we do not intend to achieve the best accuracy compared to the baselines. Instead, we propose an alternative lightweight solution to the multimodal classification task, without compromising performance. The resources related to the paper are made publicly available to facilitate reproducibility[1].

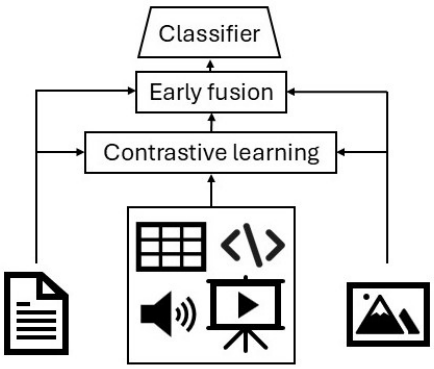

Figure 1: **Proposed Framework.** We utilize the early fusion approach, whereby embeddings from the respective modalities are concatenated and passed through a classifier network (a simple feed-forward network) to make a prediction. Exploiting the superiority of the text and image modalities, we further utilize contrastive learning to enhance the performance of other modalities (audio, video, code and tabular), which in turn further enhances the performance of the early-fusion model.

[1] https://github.com/fWern/MultimodelClassification

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 9129–9138
11–16 May 2026.

2. Related work

Multimodal fusion techniques involve the integration of feature representations from various modalities into a single representation. This approach facilitates a deeper comprehension of the data, thereby providing a more comprehensive understanding of the underlying information (Gadzicki et al., 2020b). There are three common fusion methods: Early Fusion, Intermediate Fusion, and Late Fusion.

Early Fusion fuses representations of different modalities into a single representation at the input stage (Gadzicki et al., 2020b), (Snoek et al., 2005). This approach can effectively learn cross-modal relationships from low-level features, but may struggle with capturing interactions that occur at higher levels of abstraction (Stahlschmidt et al., 2022).

Intermediate Fusion fuses representations of different modalities at intermediate stages within a model (Guarrasi et al., 2024). This method can address dimensional imbalances between modalities and thus ensures that marginal representations are of comparable size.

Late Fusion processes each feature representation of the input modalities in a separate model and then fuses them together into one representation. This fused representation is then further processed for the target task (Gadzicki et al., 2020b), (Snoek et al., 2005).

To effectively fuse different modalities into one representation, several methods can be utilized:

Concatenation: combining features from multiple modalities into a single, extended feature vector (Ngiam et al., 2011), (Zeng et al., 2019), (Zhao et al., 2024).

Element-wise Addition: adding corresponding features from different modalities to integrate the information (Zhao et al., 2024).

Element-wise Multiplication: multiplying corresponding features to emphasize interactions between them (Kim et al., 2016), (Zhao et al., 2024).

Cross Product: computing pairwise interactions between features of different modalities to create a tensor that captures complex relationships between them (Zadeh et al., 2017), (Zhao et al., 2024).

As multimodal data continue to grow in scale and complexity, traditional fusion methods can struggle to capture the intricate relationships between diverse data sources. To address these challenges, advanced fusion methods have come up, offering sophisticated techniques to enhance the integration and interpretation of multimodal information (Zhao et al., 2024). We discuss these methods in the following.

Encoder-decoder methods. (Zhao et al., 2024) In this setup, the raw or extracted features from each modality are passed as input to the encoder to obtain a latent representation, which preserves necessary semantic information. The decoder then utilizes this latent representation to make predictions. The fusion of information in the encoder could be at the raw data level (Zhao et al., 2024), hierarchical (Chen et al., 2024) or decision level (Zhao et al., 2024).

Attention methods. Attention mechanisms enable models to assign different weights to various parts of the input data. This allows the model to focus on the most relevant information for the given task, leading to better predictions without significantly increasing the computational cost. In the context of multi-modal tasks, the existing methods could be categorized into intra-modality self-attention or inter-modality cross-attention. Intra-modality self-attention allows the model to focus on different parts of a single modality, capturing dependencies and relationships between various components within that modality's features (Vaswani, 2017). By using techniques like dot-product attention (N, 2021) or additive-gate based attention (Oktay et al., 2018), this approach ensures that the model attends exclusively to intra-modality information. On the other hand, inter-modality Cross-Attention enables the model to focus on interactions between different modalities by computing attention scores across multiple modalities. This approach allows the model to align and integrate features from one modality with those from another, helping to capture dependencies and relationships across modalities (Zhao et al., 2023).

Graph neural network based. This involves creating Graph-Based embeddings for each modality. Each modality is represented as a graph where edges and nodes capture features and relationships (Lotfi et al., 2021; Qian et al., 2021).

Present work. Here we implement an early fusion approach alongside a contrastive learning method designed to enhance the performance of early fusion by improving the representation of individual modalities. By leveraging contrastive learning, the challenges of aligning and effectively combining modalities are addressed, as it helps the model learn better, more distinct representations.

3. Methodology

In this section, we first explain the architecture of the early fusion approach and then discuss ways to incorporate contrastive learning as a technique to enhance cross-modal alignment.

3.1. Early Fusion

In the first step, the features are extracted from the raw data of different modalities. For texts and im-

ages, this is done using the CLIP model (Radford et al., 2021). For tabular data, code data, and audio data, the feature extractors used are, respectively, Autogluon-Tabular (Erickson et al., 2020), CodeBERT (Feng et al., 2020) and Wav2Vec (Schneider et al., 2019). For video modality, 15 frames are extracted and processed sequentially using the CLIP image encoder, and then max-pooling is performed to obtain the final embedding. After concatenation of the features of various modalities, the resulting embedding is fed into the classification model. The classifier is made up of two processing blocks. Each block consists of several layers to sequentially refine and transform the input features for the classification task. Each block includes a fully connected layer, followed by a normalization layer, specifically employing batch normalization. After normalization, the ReLU activation function is applied to introduce non-linearity into the model. For the loss function, *Cross-Entropy* is employed, which is well-suited for multi-class classification problems (Demirkaya et al., 2020).

3.2. Enhancing Cross-modal Alignment

Building on the observation that *text* and *image* are often the dominant modality, we propose a contrastive-learning-inspired alignment method to enhance the feature quality of other modalities. Following this enhancement, these modalities can be integrated into the early fusion mechanism for improved performance. The enhancement is achieved in two steps - (i) projection and (ii) alignment through contrastive learning.

Projection: In this step, the modality selected for the enhancement is passed through a *projector* module, which includes a linear layer followed by a Tanh activation. A dropout layer is then added to mitigate overfitting. The final step involves combining the output of the dropout layer with the initial projection via a residual connection. This connection is crucial for maintaining the original features of the input, ensuring that essential information is retained. At the same time, the model learns richer and more meaningful representations through contrastive learning, which we elaborate on next.

Alignment through contrastive learning. We utilize primarily the *text* (T) and *image* (I) embedding (where applicable) to enhance another modality (say M). A learning instance is a triplet (M^i, T^i, I^i) (or a pair if the image modality is not available). Given a batch of instances, we first extract the features corresponding to the three modalities. For M, we further pass it through the projector module. Two similarity matrices are then computed—one between the text feature embedding and the projected feature embedding, and another between the image feature embedding and the projected feature embedding. Specifically, for text modality feature embedding x_T^i and feature embedding for M, x_M^j, the similarity is computed as -

$$cos\_sim_{i,j} = \exp\left(\frac{x_T^i \cdot x_M^j}{\|x_T^i\|_2 \|x_M^j\|_2}/\tau\right)$$

And similarly for image (when available). The calculated similarity matrix is scaled by a temperature factor τ and serves as the basis for distinguishing between positive and negative pairs. The positive pairs for each sample are represented by the diagonal elements of the similarity matrix, expressed as:

$$pos\_sim_i = cos\_sim_{i,i}$$

In this context, pos_sim denotes the extracted pairs from the diagonal of the matrix.
To focus the loss calculation on the hardest negative samples, a mask is created to exclude positive pairs from the set of potential negatives. By focusing on a limited number of the hardest negative pairs, the ability of the model to differentiate between similar and dissimilar inputs is enhanced. This approach is used over using all negative samples because it reduces noise and enables the model to focus on the most informative examples, helping it to learn more effectively.

Depending on the specified number of the hardest negatives to consider, the top hardest negative similarities are selected for each sample. If fewer negatives are available than the specified number, all negatives are used.

The sum of the selected hardest negative similarities is computed for each sample, which will be used in contrast with the positive similarities:

$$neg\_sum_i = \sum_{j=1}^{k} top\_k\_neg\_sim_{i,j}$$

Here, $top\_k\_neg\_sim$ represents the hardest negative similarities selected from the masked similarity matrix, comprising the k highest similarity scores from negative pairs for each sample.

Finally, a modified *NT-Xent* (Normalized Temperature-Scaled Cross Entropy) (Chen et al., 2020) loss function is used to calculate the loss. This function is defined as the negative logarithm of the ratio between the positive similarity and the combined sum of the positive similarity and the hardest negative similarities. This adaptation allows the model to better distinguish between

positive and negative pairs by focusing on the hardest negatives. Thereby improving the overall learning effect of the contrastive learning method.

Including the positive similarities in the denominator serves several purposes. It normalizes the loss with respect to the strength of positive pairs, ensuring that the model learns to emphasize the most relevant distinctions between similar and dissimilar samples. It also balances the contributions of positive and negative pairs, enhancing the model's ability to discriminate effectively among them. The loss function can be expressed as:

$$loss_i = -\log\left(\frac{pos\_sim_i}{pos\_sim_i + neg\_sum_i + \epsilon}\right)$$

where pos_sim_i represents the positive similarity for sample i, neg_sum_i is the sum of the similarity scores for the k hardest negative pairs for sample i, and ϵ is a small constant (e.g., 1×10^{-7} added to avoid division by zero). The calculated loss is then averaged over the batch to obtain the final loss value:

$$text\_loss = \frac{1}{N}\sum_{i=1}^{N} loss_i$$

where N is the amount of samples in the batch.

This process is similarly applied to the image modality to compute the corresponding contrastive loss, referred to as *image_loss*, by utilizing the image feature representations.

To obtain the overall loss for the contrastive learning process, the mean of these two losses is calculated:

$$contrastive\_loss = \frac{text\_loss + image\_loss}{2}$$

4. Experiments

4.1. Datasets

We perform experiments on datasets comprising different modalities, including audio, video, image, and code, in addition to the text modality.

4.1.1. MuG Benchmark

The MuG benchmark (Lu et al., 2023) consists of eight datasets derived from four different games: Pokémon, Hearthstone, League of Legends (LoL) and Counter Strike: Global Offensive (CS:GO). Each dataset within this benchmark presents unique classification tasks to the specific features of the respective game.

For the Pokémon dataset, the task is to predict the primary and the secondary types of Pokémon. This involves classifying Pokémon into several types.

The Hearthstone dataset is divided into four separate datasets with its own classification task: (1) Card Class: predicting the class to which a card belongs, (2) Card Set: identifying the set from which a card originates, (3) Minion Race: identifying the race of a minion card, and (4) Spell School: determine the school of magic associated with a spell card. For the LoL dataset, the task is to predict the category of a skin. The task of the CS:GO dataset is to assess the quality of in-game skins.

In each of these datasets, a sample is described with a text, an image, and tabular data. The text data in the various datasets is distributed across different columns within the tabular data.

Pre-processing. We follow the same pre-processing pipeline as in (Lu et al., 2023). It leverages the AutoGluon Multimodal package[2], which automatically identifies the types of columns within the tabular data. Specifically, columns identified as either textual or categorical are selected for text processing. The content of these selected columns is then concatenated into a single string, which is subsequently tokenized using the CLIP tokenizer for further processing. The tokenizer converts each word or character in the text sequence into a corresponding token ID using a predefined vocabulary, and returns a PyTorch tensor, which is expected as input for processing with CLIP.

Images are preprocessed by using CLIP's image preprocessing pipeline to ensure compatibility with the model. This process involves resizing and center-cropping the image to standardized dimensions and normalizing the pixel values to match the expected input range of the model. Finally, the image is converted into a tensor format, thereby preparing it for subsequent processing within the model. The resulting feature embedding for each image and text sample is of size $\mathbb{R}^{512}$. Once these features are extracted from all samples, they are stacked to form a combined feature embedding. This feature embedding has a dimension size of $\mathbb{R}^{n\times512}$ for each text and image modality, where n is the number of samples.

An automatic feature generator[3] from AutoGluon is utilized to automate the extraction of meaningful features from the tabular data. It is configured to exclude text special features such as word count, capital letter ratio, and symbol counts, n-grams, and vision based features. This enables focus solely

[2] https://pypi.org/project/autogluon.multimodal/

[3] https://auto.gluon.ai/dev/_modules/autogluon/features/generators/auto_ml_pipeline.html#AutoMLPipelineFeatureGenerator

on the core tabular data. Following feature extraction, a tabular neural network[4] is initialized. It includes processing steps to normalize and transform the extracted features into tensors suitable for input into the neural network. Each processed feature set is subsequently converted into tensors, resulting in feature embeddings with a dimension of $\mathbb{R}^{n \times m}$, where n represents the amount of rows (samples) in the table and m the number of columns (features).

Evaluation metrics. For the MuG benchmark, Accuracy and Log Loss are reported as evaluation metrics. Table 1 provides an overview of the statistics of this dataset.

Dataset	Game	Target	#Train, #Valid, #Test	#Classes
pokemon_t1	Pokémon	Primary Type	719, 45, 133	18
pokemon_t2	Pokémon	Secondary Type	719, 45, 133	19
hs_ac	Hearthstone	Cards' Class	8,561, 536, 1,603	13
hs_as	Hearthstone	Cards' Set	8,548, 532, 1,603	37
hs_mr	Hearthstone	Cards' Minion Race	5,398, 337, 1,012	15
hs_ss	Hearthstone	Cards' Spell School	2,715, 170, 508	8
lol_sc	LoL	Skin Category	1,000, 63, 188	7
csgo_sq	CS:GO	Skin Quality	766, 49, 141	6

Table 1: Statistics of the MuG Datasets

4.1.2. CMU-MOSI and CMU-MOSEI

The early fusion and contrastive learning methods are also applied to the task of sentiment classification. For this, two multimodal sentiment analysis datasets, *CMU-MOSI* (Zadeh et al., 2016) and *CMU-MOSEI* (Zadeh et al., 2018), are used. These datasets utilize text, video, and audio as modalities. *CMU-MOSI* contains 2,199 monologue utterance slices from 93 YouTube videos. The data is split into 1,284 training, 229 validation, and 686 testing utterances.

CMU-MOSEI includes 20,000 video clips derived from 3,228 videos on 250 varied topics, sourced from 1,000 unique speakers on YouTube. It utilizes 16,326 utterances for training, 1,871 utterances for validation and 4,569 utterances for testing.

Pre-processing. For both datasets, the text modality is preprocessed using the CLIP tokenizer. Given that CLIP is not designed for direct application on video data, 15 frames are extracted from each video at specific intervals, ensuring that the selection captures a comprehensive representation of the video's content. These frames are then preprocessed by the image preprocessing pipeline to prepare them for feature extraction.

The audio input files are initially loaded with a sample rate of 16 kHz and processed by the Wav2Vec processor, converting them into tensors as the required format for the model. These tensors are then fed into the Wav2Vec model, where the features are extracted from the final hidden state. The resulting feature representation for a single audio file has dimensions of $\mathbb{R}^{n \times 1024}$, where n represents the feature sequence after downsampling and 1024 the hidden size of the model.

After the feature extraction process for an individual audio file, a mean-pooling operation is applied. Using this pooling operation results in a feature embedding of dimension $\mathbb{R}^{1024}$. Finally, analogous to the text and image feature extraction, all audio files are stacked, resulting in a final feature embedding of size $\mathbb{R}^{N \times 1024}$, where N is the number of audio samples.

Evaluation metrics. To evaluate these datasets, the following metrics are reported: Seven-Class Classification Accuracy (Acc7) for sentiment classification within the range of [-3, 3], and Binary Classification Accuracy (Acc2) for distinguishing between positive and negative sentiments, along with the Weighted F1-Score. Table 2 provides an overview of train, validation, and test splits for the two datasets.

Dataset	#Train, #Valid, #Test	#Classes
CMU-MOSI (Zadeh et al., 2016)	1,284, 229, 686	2, 7
CMU-MOSEI (Zadeh et al., 2018)	16,326, 1,871, 4,659,	2, 7

Table 2: Statistics of the Sentiment Classification Datasets

4.1.3. MuIDIC Datasets

Kwak et al. (Kwak et al., 2023) present four datasets that include the modalities text, image, and code. For these datasets, they select four open-source projects available on GitHub, based on their active issue management and the presence of bug and feature labels: *VS Code*, *Kubernetes*, *Flutter* and *Roslyn*. They specifically choose projects to perform binary classification of issue reports. Each project presents unique issue characteristics: *VS Code* focuses on development environment issues, *Kubernetes* on container management, *Flutter* on mobile app development, and *Roslyn* on C# and VB.NET code analysis.

Due to significant class imbalances across all projects, the datasets were downsampled to ensure that 'bug' and 'feature' labels were balanced. The final datasets consisted of 980 samples for VS Code, 148 samples for Kubernetes, 759 samples for Flutter, and 263 samples for Roslyn. The datasets were split into training and testing sets, using an 80/20 split. Table 3 shows an overview of these splits for the different datasets.

[4] https://auto.gluon.ai/stable/_modules/autogluon/tabular/models/tabular_nn/torch/tabular_nn_torch.html#TabularNeuralNetTorchModel

Dataset	#Train, #Test	#Classes
VS Code	1,566, 394	2
Kubernetes	236, 60	2
Flutter	1,214, 304	2
Roslyn	420, 106	2

Table 3: Statistics of the MulDIC Datasets

Pre-processing. For these datasets, the text and images are preprocessed using the CLIP tokenizer and CLIP's image preprocessing pipeline.

For the code samples, CodeBERT (Feng et al., 2020) is utilized. This model is developed by Microsoft and specifically trained on programming languages. It aims to understand code by leveraging the bi-directional context of tokens. The model is built on the RoBERTa (Liu, 2019) architecture and pretrained on a large corpus of code from multiple programming languages such as Python, Java, JavaScript, and more. This makes it useful for the task of extracting features from code used in the different datasets. To preprocess the given code samples, the RoBERTa tokenizer is used. This tokenizer converts code snippets into a format suitable for CodeBERT. The tokenizer is pretrained on the same corpus as CodeBERT, thus ensuring consistency in how the code is tokenized. After tokenization, the sequences are padded to ensure that all sequences are of the same length. If any code snippet exceeds the maximum allowable length of 512 tokens, they are truncated to fit within this limit. After that, each token in a sequence is replaced by a unique integer ID that corresponds to a token in the tokenizer vocabulary. Furthermore, an attention mask is created that indicates which tokens should be attended to by the model (0 for padding tokens, 1 for real tokens). The [CLS] token added at the beginning of the sequence serves as a compact representation of the entire code snippet and can effectively summarize its semantic meaning. The resulting feature representation for a single code snippet has a dimension of $\mathbb{R}^{768}$. After extracting features from all code snippets, the features are stacked, resulting in a dimension of size $\mathbb{R}^{n\times 768}$, where n is the number of code snippets.

Evaluation metrics. For evaluation, the same metrics used by Kwak et al. (Kwak et al., 2023) are applied: weighted precision, recall, and F1-Score.

4.2. Baselines

We now elaborate on the baseline approaches that we consider for each dataset.

4.2.1. MuG

For the MuG datasets, we consider the following baselines.

MuGNet (Lu et al., 2023): a graph neural network that comprises three key components: an Adaptive Multiplex Graph Construction Module, a GAT encoder module, and an Attention-Based Fusion module.

AutoGluon (Erickson et al., 2022): an Ensemble-Learning model for multimodal classification and regression tasks.

AutoMM (Shi et al., 2021): a Late-Fusion model that uses separate neural operations on each datatype.

Method	pkm_t1	pkm_t2	hs_mr	hs_ss	csg_sq
T	0.704	0.602	0.832	0.873	0.668
I	0.480	0.498	0.843	0.782	0.691
Tab	0.456	0.495	0.556	0.687	0.695
Early fusion (T+I)	0.714	0.602	0.905	**0.889**	0.678
Early fusion (T+Tab)	0.729	0.638	0.828	0.843	0.679
Early fusion (I+Tab)	0.623	0.576	0.866	0.803	0.709
Early fusion (All)	0.755	0.665	0.902	0.874	0.722
AutoGluon	0.744	0.617	0.879	0.882	**0.766**
AutoMM	0.639	0.511	0.549	0.671	0.738
MuGNet	0.774	0.669	**0.908**	0.880	0.745
Tab^{CL}	0.550	0.577	0.575	0.673	0.711
T + Tab^{CL}	0.734	0.671	0.825	0.836	0.714
I + Tab^{CL}	0.669	0.630	0.868	0.782	0.705
Early fusion^{CL}	**0.785**	**0.687**	0.902	0.876	0.735

Table 4: **Performance comparison MuG benchmark.** We report the accuracies across different tasks within the benchmark. Note that early fusion considering all the modalities often achieves performance comparable to the baseline methods. Text modality individually often achieves superior performance compared to other modalities. We further enhance the Tab modality using contrastive learning (Tab^{CL}), which further improves the performance when used individually or incorporated into early fusion with other modalities.

4.2.2. MulDIC Datasets

We consider - **MulDIC** (Kwak et al., 2023), a multimodal classification model based on CNNs for three modalities: text, image, and code.

4.2.3. Sentiment classification

For sentiment classification, we consider -

MissModal (Lin and Hu, 2023), a multimodal sentiment analysis approach that aims to increase robustness to missing modalities by aligning modal-missing and modal-complete data.

MAG-BERT (Rahman et al., 2020), which incorporates a Multimodal Adaptation Gate (MAG) to extend BERT for multimodal language tasks.

MISA (Hazarika et al., 2020), a multimodal affective framework that factorizes modalities into Modality-Invariant and Modality-Specific features and then fuses them to predict the sentiments.

	VS Code			Kubernetes			Flutter			Roslyn		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
T	0.731	0.729	0.729	0.749	0.737	0.733	0.651	0.649	0.649	0.669	0.668	0.668
I	0.555	0.554	0.552	0.662	0.610	0.575	0.540	0.540	0.539	0.582	0.581	0.581
C	0.583	0.579	0.575	0.751	0.727	0.720	0.600	0.550	0.493	0.585	0.581	0.575
Early fusion (T+I)	0.715	0.710	0.708	0.718	0.690	0.677	0.667	0.665	0.663	0.717	0.717	0.717
Early fusion (T+C)	0.719	0.713	0.711	0.834	0.830	0.830	0.696	0.679	0.672	0.700	0.698	0.697
Early fusion (I+C)	0.567	0.564	0.559	0.728	0.713	0.709	0.572	0.564	0.555	0.605	0.604	0.603
Early fusion (All)	0.717	0.712	0.711	0.840	0.827	0.825	0.679	0.668	0.672	0.717	0.709	0.709
MulDIC	**0.804**	**0.797**	**0.801**	0.813	0.800	0.806	**0.769**	**0.757**	**0.763**	0.680	0.679	0.679
C^{CL}	0.579	0.577	0.574	0.763	0.757	0.755	0.582	0.574	0.564	0.591	0.591	0.590
T+C^{CL}	0.718	0.710	0.709	0.828	0.827	0.827	0.685	0.678	0.675	0.697	0.696	0.686
I+C^{CL}	0.577	0.576	0.575	0.801	0.767	0.758	0.564	0.561	0.556	0.599	0.598	0.597
Early fusion^{CL}	0.728	0.721	0.719	**0.843**	**0.837**	**0.836**	0.692	0.682	0.692	**0.735**	**0.734**	**0.734**

Table 5: **Performance comparison on MulDIC.** We report the accuracies across different tasks within the benchmark. Note that early fusion considering all the modalities often achieves performance comparable to the baseline methods. Text modality individually often achieves superior performance compared to other modalities. We further enhance the Code modality through contrastive learning (C^{CL}), which enhances its individual performance as well as when incorporated with other modalities in early fusion fot *kubernetes* and *Roslyn*.

5. Results

For all the datasets and tasks, we only consider early fusion, which involves extracting features from each modality, concatenating them and training a classifier model on top.

5.1. Performance of individual modalities

We start by investigating the performance of individual modalities. Specifically, we consider each modality individually by obtaining the corresponding features utilizing the pre-processing step outlined previously, followed by a classifier layer on top. The results are presented in tables 4, 5 and 2 respectively for MuG, MulDIC and sentiment classification tasks respectively. Notably, text modality individually achieves comparable performances to the baseline methods that involve all the modalities. The other modalities are not observed to be so effective. For example, for *pkm_t1* task the text (T) modality individually achieves an accuracy of 0.704 while the image (I) and tabular (Tab) only achieve accuracies of 0.48 and 0.45 respectively, while the best-performing baseline involving all modalities achieves an accuracy of 0.77.

5.2. Effectiveness of early fusion

We next demonstrate that early fusion often achieves performance comparable to more complex approaches. Specifically, we extract features from the respective modalities, concatenate them and train a classifier model on top. Experiments are performed considering all the modalities or a subset. On the MuG benchmark (table 4), simple early fusion involving all the modalities (Early fusion (All)) achieves performance similar to the most competitive baseline with a fraction of time and parameter complexity. For MulDIC benchmark (table 5), early fusion with all modalities even outperforms the baseline in Kubernetes and Roslyn datasets. The performance is worse for VS Code and Flutter datasets. Similarly, for the sentiment classification task, early fusion achieves competitive performance for the CMU-MOSEI dataset. These results together demonstrate that early fusion could be a simple yet effective alternative for different classification tasks involving a variety of modalities.

5.3. Enhancing Modalities

Utilizing the superiority of text features, we now deploy contrastive learning to enhance the performance of other modalities. For the MuG benchmark, the tabular modality is the one considered for enhancement. Table 4 presents the performance of the tabular modality individually as well as in combination with the early fusion approach. Notably, the performance of the tabular modality improves significantly for *pkm_t1* and *pkm_t2*, while the improvement is marginal for the others. This further results in improved performance for the early fusion model, which outperforms the baselines in these two datasets. Combining the contrastively learned tabular features with other modalities also enhances the performance.

For the MulDIC benchmark, the code modality is considered for enhancement. The results (table 5) indicate that contrastive learning can lead to an enhanced performance on specific modality combinations across the datasets. These improvements are observed almost only in the code modality or in the combination of the image and code modalities. The contrastive setup allows for further improvement in performance in early fusion experiments,

Method	CMU-MOSI		CMU-MOSEI	
	Acc	F1	Acc	F1
T	0.724	0.724	0.818	0.815
V	0.576	0.574	0.736	0.720
A	0.589	0.588	0.720	0.629
Early fusion (T+V)	0.713	0.713	0.812	0.808
Early fusion (T+A)	0.743	0.744	0.813	0.810
Early fusion (V+A)	0.545	0.544	0.735	0.707
Early-fusion (All)	0.719	0.719	0.818	0.815
MISA	0.818	0.817	0.819	0.820
MAG-BERT	0.824	0.822	0.819	0.823
MissModal	0.841	0.840	0.834	0.836
A^{CL}	0.526	0.482	0.728	0.657
T + A^{CL}	0.747	0.747	0.811	0.807
T + V^{CL}	0.573	0.573	0.744	0.730
Early-fusion^{CL}	0.727	0.727	0.818	0.815

Table 6: **Performance comparison in multimodal sentiment classification.** We report the accuracies across different tasks within the benchmark. Note that early fusion, considering all the modalities, often achieves performance comparable to the baseline methods. We also train classifier models considering individual and combinations of text (T), video (V) and audio (A) modalities. Text modality often individually achieves superior performance compared to other modalities.

with the best performance obtained for *Kubernetes* and *Roslyn* datasets.

The results on the *CMU-MOSI* and *CMU-MOSEI* datasets are presented in table 6. Both audio and video modalities are considered for enhancement. Only marginal improvements are observed with such enhancements.

5.4. Parameter efficiency

The early fusion method requires training with much fewer parameters than the more complicated baseline methods while achieving comparable performance. For the MuG benchmark (*pkm_t1* task), AutoMM requires training $\sim 365M$ parameters, whereas early fusion with contrastive learning requires $\sim 1M$ parameters for training. MugNet requires a slightly larger number of trainable parameters than early fusion with CL. The early fusion method without contrastive learning requires only $\sim 500K$ trainable parameters. We present the performance versus number of parameters for the different classification models in figure 2. The results clearly demonstrate that early fusion variants achieve comparable performance with only a fraction of the number of parameters of the more complex baseline methods.

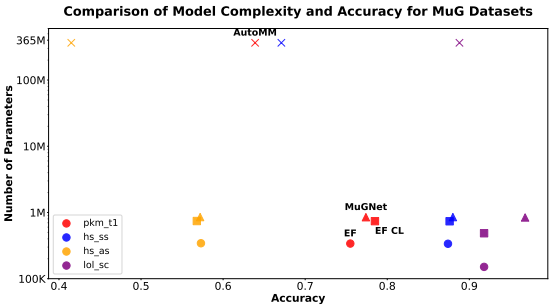

Figure 2: Comparison of the number of parameters of the different classification methods for MuGNet datasets. EF and EF CL refer to early fusion and early fusion with contrastive learning respectively.

Discussion

Implication of results. Our results demonstrate that text modality seem to be the most dominant modality. It is often the case that a model trained with any other modality in combination with the text modality relies more on the text modality when performing inference. One must ensure that the model can effectively utilize other modalities for more robust results.

The effectiveness of the early fusion method, as demonstrated by the results, suggests it could be a good starting point for designing multimodal architectures for a given task. Complexity should only be introduced if early fusion fails to align different modalities effectively, and that modality alignment is a critical requirement for the task.

Limitations. For our text and image modalities, we only considered embeddings from CLIP text and image encoders. More advanced text and vision encoders could be used to improve performance.

While the proposed early fusion method achieves comparable results on several tasks, the improvements are not so pronounced in more complex tasks (e.g., multimodal sentiment classification).

Our contrastive learning setup relies on the quality of the text embeddings (additionally, image embeddings, when available). However, if the text embedding is noisy or missing, which might be encountered in a real-world application scenario, our setup might not be that useful. In such cases, a separate component for noise reduction should be incorporated into the framework.

6. Conclusion

We explored the multimodal classification task, implementing an Early Fusion approach alongside a Contrastive Learning method. Through comprehensive evaluation across three genres of datasets, we observe that our proposed framework achieves accuracy comparable to most competitive state-of-

the-art models, with significantly lower complexity in terms of the number of trainable parameters. Notably, increasing model complexity does not necessarily lead to better performance, and simpler models can offer advantages in terms of efficiency and generalization.

7. Acknowledgments

This research was in part supported by the Federal Ministry of Education and Research by the Lower Saxony Ministry of Science and Culture (MWK) through the zukunft.niedersachsen program of the Volkswagen Foundation (HybrInt).

8. Bibliographical References

Haojie Chen, Zhuo Wang, Hongde Qin, and Xiaokai Mu. 2024. Dhfnet: Decoupled hierarchical fusion network for rgb-t dense prediction tasks. *Neurocomputing*, 583:127594.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ahmet Demirkaya, Jiasi Chen, and Samet Oymak. 2020. Exploring the role of loss functions in multiclass classification. In *2020 54th annual conference on information sciences and systems (ciss)*, pages 1–5. IEEE.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*.

Nick Erickson, Xingjian Shi, James Sharpnack, and Alexander Smola. 2022. Multimodal automl for image, text and tabular data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4786–4787.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, et al. 2020. Codebert: A pre-trained model for programming and natural languages. *arXiv preprint arXiv:2002.08155*.

Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. 2020a. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd international conference on information fusion (FUSION)*, pages 1–6. IEEE.

Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. 2020b. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–6.

Valerio Guarrasi, Fatih Aksu, Camillo Maria Caruso, Francesco Di Feola, Aurora Rofena, Filippo Ruffini, and Paolo Soda. 2024. A systematic review of intermediate fusion in multimodal deep learning for biomedical applications. *arXiv preprint arXiv:2408.02686*.

Devamanyu Hazarika, Roger Zimmermann, and Soujanya Poria. 2020. Misa: Modality-invariant and-specific representations for multimodal sentiment analysis. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1122–1131.

Douwe Kiela, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2018. Efficient large-scale multimodal classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jin-Hwa Kim, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2016. Multimodal residual learning for visual qa. *Advances in neural information processing systems*, 29.

Changwon Kwak, Pilsu Jung, and Seonah Lee. 2023. A multimodal deep learning model using text, image, and code data for improving issue classification tasks. *Applied Sciences*, 13(16):9456.

Ronghao Lin and Haifeng Hu. 2023. Missmodal: Increasing robustness to missing modality in multimodal sentiment analysis. *Transactions of the Association for Computational Linguistics*, 11:1686–1702.

Y Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Serveh Lotfi, Mitra Mirzarezaee, Mehdi Hosseinzadeh, and Vahid Seydi. 2021. Detection of rumor conversations in twitter using graph convolutional networks. *Applied Intelligence*, 51:4774–4787.

Jiaying Lu, Yongchen Qian, Shifan Zhao, Yuanzhe Xi, and Carl Yang. 2023. Mug: A multimodal classification benchmark on game data with tabular, textual, and visual fields. *arXiv preprint arXiv:2302.02978*.

Krishna D N. 2021. Using large pre-trained models with cross-modal attention for multi-modal emotion recognition.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. 2018. Attention u-net: Learning where to look for the pancreas.

Shengsheng Qian, Jun Hu, Quan Fang, and Changsheng Xu. 2021. Knowledge-aware multi-modal adaptive graph convolutional networks for fake news detection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(3):1–23.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, Amir Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. 2020. Integrating multimodal information in large pretrained transformers. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2020, page 2359. NIH Public Access.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *Proc. Interspeech 2019*, pages 3465–3469.

Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alex Smola. 2021. Multimodal automl on structured tables with text fields. In *8th ICML Workshop on Automated Machine Learning (AutoML)*.

Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, page 399–402, New York, NY, USA. Association for Computing Machinery.

Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. 2022. Multimodal deep learning for biomedical data fusion: a review. *Briefings in Bioinformatics*, 23(2):bbab569.

Ashish Vaswani. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Weiyao Wang, Du Tran, and Matt Feiszli. 2020. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12695–12705.

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246.

Jin Zeng, Yanfeng Tong, Yunmu Huang, Qiong Yan, Wenxiu Sun, Jing Chen, and Yongtian Wang. 2019. Deep surface normal estimation with hierarchical rgb-d fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6153–6162.

Fei Zhao, Chengcui Zhang, and Baocheng Geng. 2024. Deep multimodal data fusion. *ACM Computing Surveys*, 56(9):1–36.

Qiankun Zhao, Yingcai Wan, Jiqian Xu, and Lijin Fang. 2023. Cross-modal attention fusion network for rgb-d semantic segmentation. *Neurocomputing*, 548:126389.