Thinking Nodes at MAHED: A Comparative Study of Multimodal Architectures for Arabic Hateful Meme Detection

Itbaan Safwan
Institute of Business Administration
Karachi, Pakistan
i.safwan.26197@khi.iba.edu.pk

Abstract

This paper describes our system for Task 3 of the Arabic NLP 2025 competition: detecting hateful content in Arabic memes. The task requires a robust understanding of both visual and textual information and their interplay. We developed and compared three distinct multimodal fusion architectures: a Cross-Attention model, a progressive CNN-based fusion model, and a two-stage model using custom-trained embeddings with a gated fusion classifier. All models leverage pre-trained CLIP and MARBERT encoders for image and text representation, respectively. We detail our approach to handling the significant class imbalance in the dataset through data re-splitting and the application of a weighted Focal Loss. Our post-competition analysis, training on all available data, shows that the CNN-based fusion model achieved the highest macro F1-score of 0.779, demonstrating the effectiveness of its hierarchical feature extraction for this task.

1 Introduction

The proliferation of memes on social media has transformed them into a potent medium for communication, but also for the spread of hate speech. Detecting hateful content within memes is a challenging multimodal task, as the malicious intent often arises not from the image or text in isolation, but from their complex and often ironic interplay. This paper presents our contribution to the Arabic NLP 2025 Shared Task 3 on Multimodal Hateful Meme Detection (Zaghouani et al., 2025), which focuses on classifying Arabic memes as hateful or not hateful.

Previous work has established benchmarks for multimodal hate speech detection, often focusing on English memes and exploring various fusion strategies (Kiela et al., 2021). While recent efforts have begun to build valuable resources for Arabic, such as the ArMeme dataset (Alam et al., 2024b), a systematic comparison of different deep fusion architectures specifically for hateful Arabic memes remains an area ripe for exploration. The optimal way to combine visual and textual cues—whether by capturing global context or local patterns—is not yet well understood for this specific domain.

To address this gap, we conduct a comparative analysis of three distinct fusion architectures, leveraging powerful pre-trained CLIP and MARBERT encoders as our backbones. We investigate a global Cross-Attention mechanism, a localized progressive CNN-based approach, and a two-stage Custom Embedding model. A key part of our methodology was also addressing the severe class imbalance in the dataset through stratified re-splitting and a weighted Focal Loss. Our experiments reveal that the progressive CNN model achieves the highest performance, demonstrating the effectiveness of learning hierarchical local features for this task.

The main contributions of this paper are as follows:

1. We provide a direct, empirical comparison of three different multimodal fusion strategies (Cross-Attention, CNN, and a two-stage contrastive approach) on the task of Arabic hateful meme detection.

2. We demonstrate an effective methodology for mitigating severe class imbalance through a combination of stratified data splitting and a weighted Focal Loss function.

3. Our post-competition analysis provides a strong performance benchmark, with our best model achieving a macro F1-score of 0.779 and highlighting the superiority of the CNN-based fusion approach for this specific task.

Our code is available at a public repository[1]

[1] https://github.com/itbaans/ArabicNLP-2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 712–719

2 Related Work

Our research is situated at the intersection of multimodal machine learning, hate speech detection, and Arabic Natural Language Processing. This section reviews key advancements in these areas to contextualize our contributions.

2.1 Multimodal Hate Speech Detection

The task of identifying hate speech has expanded from text-only analysis to the more complex domain of multimodal content. The Hateful Memes Challenge by Kiela et al. (2021) was a seminal work that established a benchmark for the task, highlighting cases where models fail if they cannot reason jointly about the image and text. Early approaches often relied on simple fusion, such as concatenating features from separate unimodal encoders. More recent works have focused on developing sophisticated deep fusion mechanisms. Cross-attentional models, which learn to align and integrate features from different modalities, have shown strong performance in various vision-and-language tasks and have been widely adopted for meme analysis (Tan and Bansal, 2019). Our work contributes to this line of research by directly comparing a cross-attention architecture with alternative fusion strategies.

2.2 Arabic Multimodal and Hate Speech Resources

While multimodal research has historically been dominated by English-language resources, there has been a significant and growing effort to develop datasets and models for Arabic. For text-based hate speech, Zaghouani et al. (2024) provided a large, richly annotated dataset of Arabic tweets, demonstrating the effectiveness of transformer-based models like AraBERT for the task. The challenge of multimodality in Arabic memes has been tackled more recently. Alam et al. (2024b) introduced ArMeme, the first major dataset for multimodal analysis of Arabic memes, providing annotations for various tasks including propaganda detection. Building on this, Alam et al. (2024a) explored the critical intersection between propaganda and hate speech in memes, using a multi-agent LLM approach to annotate and analyze this relationship. Concurrently, efforts like the ArAIEval shared task have spurred research into multimodal propaganda detection, with participants such as Shah et al. (2024) successfully employing fusion architectures combining BERT with vision models like ConvNeXt.

Our work builds directly on these foundational efforts. While previous studies have focused on creating resources or detecting propaganda, our paper provides a focused, comparative study of different deep fusion architectures specifically for the nuanced task of hate speech detection in Arabic memes, using the dataset provided by the ArabicNLP 2025 shared task.

3 System Overview

To conduct our comparative analysis, we developed three distinct multimodal architectures. All models share a common foundation, utilizing powerful pre-trained encoders for initial feature representation, but differ significantly in their strategy for fusing these features. A detailed breakdown of each model's architecture, including layer configurations and hyperparameters, is available in Appendix A.

3.1 Backbone Encoders

For visual feature extraction, we employ the vision transformer from **openai/clip-vit-base-patch32** (Radford et al., 2021). For the corresponding Arabic captions, we use **UBC-NLP/MARBERT** (Abdul-Mageed et al., 2021). In our end-to-end models, we adopt a partial fine-tuning strategy, unfreezing only the final two layers of each encoder to adapt them to the specific domain of Arabic memes while preserving their rich, general-purpose knowledge.

3.2 Fusion Architectures

Model 1: Cross-Attention Fusion This model (Figure 1) is designed to capture the global, interdependent context between modalities. Inspired by co-attentional transformers (Tan and Bansal, 2019), it uses a bidirectional cross-attention mechanism where image and text features query each other to form contextually enriched representations before being pooled and classified.

Model 2: CNN-based Fusion In contrast, this architecture (Figure 2) aims to learn localized, compositional features. Motivated by the effectiveness of convolutions for fusing aligned sequences (Zadeh et al., 2017), this model uses a stack of 1D convolutional layers to progressively fuse the image and text embedding sequences, allowing it to

build a hierarchical understanding of their interaction.

Model 3: Custom Embedding Fusion This model (Figure 3) follows a two-stage pipeline to decouple modality alignment from classification. In the first stage, we pre-train a custom dual-encoder model using a contrastive loss, following the CLIP methodology (Radford et al., 2021), to align the image and text features into a shared embedding space. In the second stage, a lightweight classifier fuses these pre-computed embeddings using a gated mechanism (Arevalo et al., 2017), which dynamically weights the contribution of each modality for the final prediction.

4 Experimental Setup

4.1 Dataset and Preprocessing

The original dataset, introduced by Zaghouani et al. (2024) and analyzed for multimodal hate speech by Alam et al. (2024a), was provided with separate train, development, and test splits. We observed a significant class imbalance, particularly in the development set, which could skew validation performance. To create a more stable training and evaluation environment, we combined all provided labeled data (train, dev, and the labeled test set from a previous phase) and performed a new stratified split, allocating 70% for training and 30% for validation. This ensured that the class proportions were consistent across both splits.

4.2 Handling Class Imbalance

The dataset is heavily skewed towards the 'not-hate' class. To mitigate this, we employed a weighted Focal Loss (Lin et al., 2018) instead of standard cross-entropy. Focal Loss addresses class imbalance by down-weighting the loss assigned to well-classified examples, thereby focusing training on hard, misclassified examples. It is defined as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \quad (1)$$

We set the focusing parameter $\gamma = 2$. The balancing parameter α_t was set using class weights computed inversely proportional to class frequencies:

$$w_c = \frac{N}{2 \times N_c} \quad (2)$$

where N is the total number of samples, and N_c is the number of samples in class c. These weights were passed to the loss function, increasing the penalty for misclassifying the minority 'hate' class.

4.3 Implementation Details

All models were trained using the AdamW optimizer with a weight decay of 1×10^{-5}. For the end-to-end models (Cross-Attention, CNN), we used a learning rate of 2×10^{-5}. For the lightweight fusion classifier (Custom Embedding), we used a higher learning rate of 5×10^{-5}. All experiments were run with a batch size of 32. We used a 'ReduceLROnPlateau' scheduler to decrease the learning rate if the validation F1-score did not improve for 2 epochs. Early stopping was implemented with a patience of 10-15 epochs to prevent overfitting.

5 Results and Analysis

We report two sets of results: pre-submission results based on models trained only on our 70% training split, and post-submission results where models were trained on the full combined dataset (train + validation) and evaluated on the official test set with gold labels. The official evaluation metric is macro F1-score.

5.1 Pre-Submission Results

For the official competition submission, we inadvertently trained our models only on our 70% training split, not the full available labeled data. The CNN and Cross-Attention models were submitted to the leaderboard. Due to time constraints, the Custom Embedding model was not submitted, but we report its projected score on the test set for comparison. Table 1 summarizes these findings.

The CNN model achieved the highest F1-score on our validation set, but both submitted models performed almost identically on the official test set. The Custom Embedding model, despite its lower validation score, shows a strong projected test score, indicating its potential.

5.2 Post-Submission Analysis

After the competition, the test set gold labels were released. This allowed us to conduct a more thorough analysis by training our models on all available labeled data (our 70% train + 30% validation splits combined) and evaluating on the official test set. The results are shown in Table 2.

Impact of Training Data Size A key finding is the significant performance boost observed across all models when trained on the full dataset versus the partial split. The CNN Fusion model's F1-score, for instance, jumped from 0.718 to 0.779 (+6.1 points). This highlights that our models were

Model	Validation Set (Our Split)				Official Test Set			
	Val F1	Precision	Recall	Accuracy	Official F1	Precision	Recall	Accuracy
Cross-Attention	0.692	0.668	0.750	0.824	0.719	0.733	0.714	0.740
CNN Fusion	**0.727**	**0.696**	**0.802**	0.840	0.718	**0.776**	0.711	**0.754**
Custom Emb.	0.690	0.683	0.698	**0.853**	**0.720***	0.752	**0.713**	0.748

Table 1: Pre-submission results. Models were trained on a 70% split of the data. Metrics for the validation set are macro-averaged for F1, Precision, and Recall. Official Test F1 is from the CodaLab leaderboard or our projection based on gold labels (*).

Model	Test F1 (Full Data)
Cross-Attention	0.765
CNN Fusion	**0.779**
Custom Emb. Fusion	0.765

Table 2: Post-submission results. Models were trained on all available labeled data and evaluated on the official test set.

data-hungry and that leveraging all available annotations was critical for achieving optimal performance. Our pre-submission results were therefore limited by our experimental oversight.

Model Comparison In the post-submission setting, the CNN Fusion model emerged as the clear top performer. Its ability to extract and fuse localized features through convolutions appears to be more effective for this task than the global context mixing of cross-attention. The progressive nature of the fusion may also allow it to build more robust cross-modal representations. The Cross-Attention and Custom Embedding models achieved identical, strong scores, demonstrating their viability, but were ultimately outperformed by the CNN-based approach. The two-stage custom embedding approach is particularly noteworthy for its efficiency at inference time, as it only requires running a very small classifier once embeddings are pre-computed.

6 Conclusion

In this paper, we presented a comparative study of three distinct multimodal architectures—Cross-Attention, progressive CNN, and a two-stage Custom Embedding fusion—for the task of Arabic hateful meme detection. Our investigation confirmed that leveraging powerful pre-trained encoders like CLIP and MARBERT provides a strong foundation. Our findings underscore two critical aspects for this task: first, the necessity of robust techniques like weighted Focal Loss to handle severe class imbalance, and second, the significant impact of training data volume on final performance. Our post-submission analysis identified the progressive CNN-based fusion architecture as the most effective, achieving a final macro F1-score of 0.779 and suggesting that learning localized, hierarchical cross-modal interactions is a particularly robust strategy for this domain.

6.1 Limitations and Future Work

Despite these promising results, our study has several limitations. A primary concern is the models' propensity to overfit, evidenced by a decline in validation performance even as training loss decreased. This suggests that the complex architectures may have memorized spurious correlations from the relatively small dataset rather than learning generalizable features of hate speech. Another key limitation is the "black-box" nature of our fusion mechanisms, which hinders the interpretability required for reliable real-world moderation systems. Furthermore, our models do not explicitly process text embedded within images, a common feature in memes.

Future work should directly address these issues. A promising direction to mitigate both data scarcity and overfitting is to employ knowledge distillation (Hinton et al., 2015). One could leverage a powerful Vision-Language Model (VLM), such as those from the CLIP or BLIP families (Radford et al., 2021; Li et al., 2022), as a "teacher" to generate a large, pseudo-labeled dataset with soft probability distributions. A more compact "student" model, like our CNN architecture, could then be trained to mimic the teacher's nuanced outputs, transferring its reasoning capabilities into a more efficient and robust model. To improve interpretability, future research could focus on generating saliency maps to highlight which image regions and text tokens most influence a prediction, providing a clearer view into the model's decision-making process.

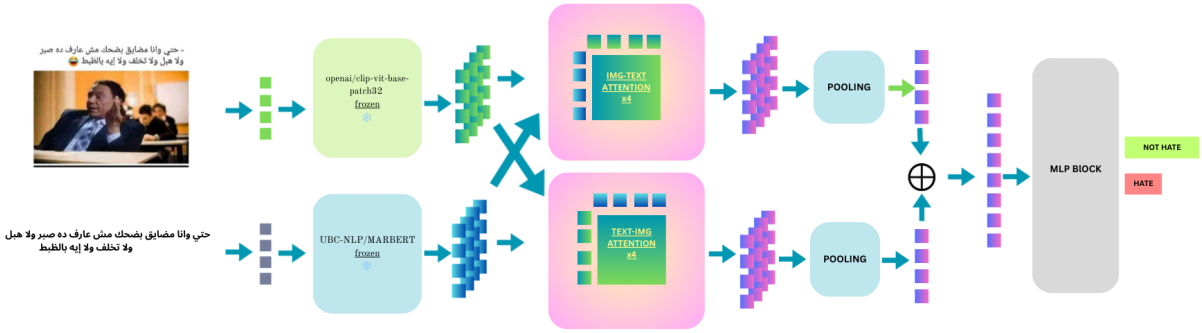

Figure 1: Architecture of the Cross-Attention Fusion model.

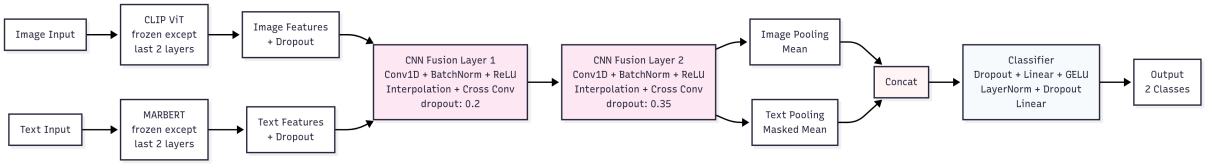

Figure 2: Architecture of the progressive CNN-based Fusion model.

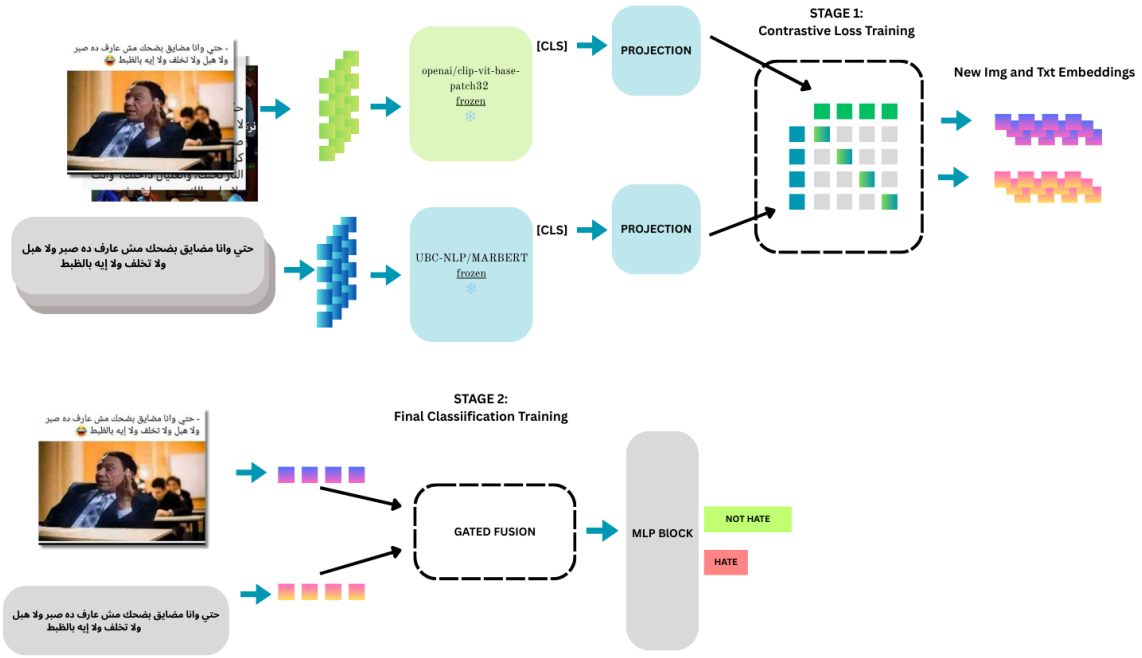

Figure 3: Architecture of the two-stage Custom Embedding Fusion model.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmad, Md. Arid Hasan, and Maram Hasanain. 2024b. ArMeme: Propagandistic content in Arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21071–21090, Miami, Florida, USA. Association for Computational Linguistics.

John Arevalo, Thamar Solorio, Manuel Montes y Gómez, and Fabio A. González. 2017. Gated multimodal units for information fusion. *Preprint*, arXiv:1702.01992.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2021. The hateful memes challenge: Detecting hate speech in multimodal memes. *Preprint*, arXiv:2005.04790.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Preprint*, arXiv:2201.12086.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Uzair Shah, Md. Rafiul Biswas, Marco Agus, Mowafa Househ, and Wajdi Zaghouani. 2024. MemeMind at ArAIEval shared task: Generative augmentation and feature fusion for multimodal propaganda detection in Arabic memes through advanced language and vision models. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 467–472, Bangkok, Thailand. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *Preprint*, arXiv:1908.07490.

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

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (Arabic-NLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

A Model Architectures and Implementation Details

A.1 CNN-Based Multimodal Fusion Model

The CNN-based fusion model (CNNMultiModalModel) employs 1D convolutional layers to process and fuse multimodal embeddings from CLIP-ViT and MARBERT encoders.

A.1.1 CNNFusionLayer Components

The core fusion component uses 1D convolutions for cross-modal interaction:

- **Input Processing**: Separate 1D convolutions for image and text embeddings with kernel size 3
- **Cross-Modal Fusion**: Concatenation followed by 1×1 convolution for dimensionality reduction
- **Normalization**: BatchNorm1d without affine parameters to prevent overfitting
- **Regularization**: Progressive dropout rates (0.2 + layer_index × 0.15)

A.1.2 Backbone Configuration

- **Vision Encoder**: CLIP-ViT-Base-Patch32 (768-dimensional embeddings)
- **Text Encoder**: MARBERT (768-dimensional embeddings)
- **Selective Unfreezing**: Only the last 2 layers of each encoder are trainable
- **Regularization**: 0.3 dropout applied to backbone outputs

A.1.3 Classification Head

The final classification component consists of:

$$\begin{aligned}\text{Classifier} = \text{Sequential}(&\\ &\text{Dropout}(0.5),\\ &\text{Linear}(\text{final\_dim} \times 2, \text{final\_dim}),\\ &\text{GELU}(),\\ &\text{LayerNorm}(\text{final\_dim}),\\ &\text{Dropout}(0.4),\\ &\text{Linear}(\text{final\_dim}, 2))\end{aligned}$$

A.2 Cross-Attention Fusion Model

The Advanced Fusion Model (AdvancedFusionModel) utilizes multi-head cross-attention mechanisms to enable bidirectional information exchange between visual and textual modalities.

A.2.1 CrossAttentionFusion Module

The fusion mechanism implements bidirectional cross-attention:

- **Text-to-Image Attention**: $\text{Att}_{t2i} = \text{MultiHeadAttn}(Q = I, K = T, V = T)$
- **Image-to-Text Attention**: $\text{Att}_{i2t} = \text{MultiHeadAttn}(Q = T, K = I, V = I)$
- **Pooling Strategies**: Support for mean, max, and learnable attention pooling
- **Feature Concatenation**: Final fusion via concatenation of pooled representations

A.2.2 Attention Pooling Mechanism

For attention-based pooling, learnable query vectors are employed:

$$\text{pooled\_img} = \text{Attention}(Q = q_{\text{img}}, K = \text{Att}_{t2i}, V = \text{Att}_{t2i}) \tag{3}$$

$$\text{pooled\_txt} = \text{Attention}(Q = q_{\text{txt}}, K = \text{Att}_{i2t}, V = \text{Att}_{i2t}) \tag{4}$$

where q_{img} and q_{txt} are randomly initialized learnable parameters.

A.2.3 Model Configuration

- **Attention Heads**: 4 heads for cross-attention modules
- **Frozen Backbones**: Complete freezing of CLIP-ViT and MARBERT parameters
- **Projection Layer**: 512-dimensional intermediate representation
- **Dropout Rates**: 0.4 for projection layer, 0.2 for classification head

A.3 Custom CLIP-Arabic with Embeddings Fusion

The custom approach involves pre-training a CLIP-style model on Arabic multimodal data, followed by embedding-based classification using various fusion strategies.

A.3.1 CLIPArabic Pre-training

The custom CLIP model implements contrastive learning:

- **Image Encoder**: Frozen CLIP-ViT-Base-Patch32
- **Text Encoder**: Frozen MARBERT

- **Projection Heads**: Linear layers mapping to 512-dimensional space
- **Contrastive Loss**: Symmetric cross-entropy on image-text similarity matrix

The contrastive loss function is defined as:

$$\mathcal{L}_{\text{contrastive}} = \frac{1}{2}(\mathcal{L}_{i2t} + \mathcal{L}_{t2i}) \tag{5}$$

$$\text{where} \quad \mathcal{L}_{i2t} = \text{CrossEntropy}(\tau \cdot \mathbf{I}\mathbf{T}^T, \mathbf{y}) \tag{6}$$

$$\mathcal{L}_{t2i} = \text{CrossEntropy}(\tau \cdot \mathbf{T}\mathbf{I}^T, \mathbf{y}) \tag{7}$$

with τ being the learnable temperature parameter and $\mathbf{y}$ the identity matrix labels.

A.3.2 Embeddings-Based Classification

The PrecomputedEmbeddingsClassifier supports multiple fusion strategies:

Gated Fusion (Best Performing):

$$\begin{aligned} g_{\text{img}} &= \sigma(W_{g,i}\mathbf{e}_{\text{img}} + b_{g,i}) \\ g_{\text{txt}} &= \sigma(W_{g,t}\mathbf{e}_{\text{txt}} + b_{g,t}) \\ \mathbf{h}_{\text{fused}} &= g_{\text{img}} \odot \text{ReLU}(W_i\mathbf{e}_{\text{img}}) \\ &\quad + g_{\text{txt}} \odot \text{ReLU}(W_t\mathbf{e}_{\text{txt}}) \end{aligned}$$

Alternative Fusion Methods:

- **Concatenation**: $\mathbf{h}_{\text{fused}} = [\mathbf{e}_{\text{img}}; \mathbf{e}_{\text{txt}}]$
- **Element-wise Addition**: $\mathbf{h}_{\text{fused}} = W_i\mathbf{e}_{\text{img}} + W_t\mathbf{e}_{\text{txt}}$
- **Element-wise Multiplication**: $\mathbf{h}_{\text{fused}} = W_i\mathbf{e}_{\text{img}} \odot W_t\mathbf{e}_{\text{txt}}$

A.4 Training Configuration and Hyperparameters

Table 3: Training hyperparameters for all models

Parameter	CNN	Cross-Attn	Custom CLIP
Learning Rate	2×10^{-5}	2×10^{-5}	5×10^{-5}
Batch Size	32	32	32
Max Epochs	30	30	30
Early Stop Patience	10	10	10
Weight Decay	1×10^{-5}	1×10^{-5}	1×10^{-5}
Gradient Clipping	1.0	1.0	1.0
Loss Function	Focal	Focal	Focal
Scheduler	ReduceLR	ReduceLR	ReduceLR

A.4.1 Focal Loss Configuration

All models employ Focal Loss to address class imbalance:

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{8}$$

where $\gamma = 2.0$ and α_t are computed based on inverse class frequencies.

A.5 Model Training Curves

To further illustrate the overfitting behavior discussed in the Limitations section, Figure 4 shows the training loss and test macro F1-score progression for all three models. In each case, the test F1-score (solid lines) peaks relatively early in training, after which it either stagnates or degrades, even as the training loss (dashed lines) continues to decrease. This divergence is a clear indicator that the models began to memorize the training data rather than learning generalizable patterns.

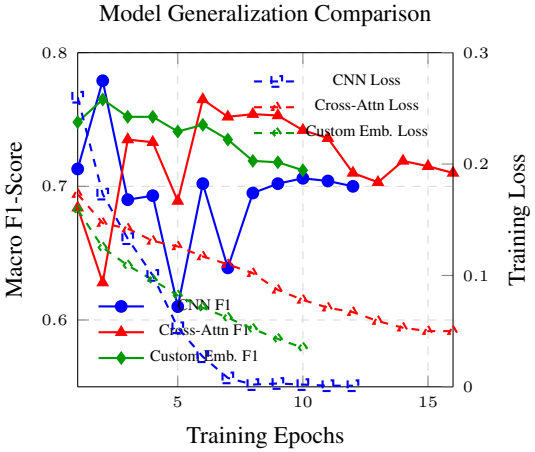

Figure 4: Comparison of training loss (dashed lines, right axis) vs. test macro F1-score (solid lines, left axis) for all models.